Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра програмного забезпечення

Пояснювальна записка

до магістерської кваліфікаційної роботи
магістр
(ступінь вищої освіти)

на тему: **Розробка методу і програмних засобів аналізу тональності тексту з використанням нейронних мереж та веб-технологій**

Студента 2 курсу, групи 1ПІ-18м (д/ф)
Спеціальності
121 «Інженерія програмного забезпечення»
Кобисі І. В.
(прізвище та ініціали)
Керівник к.т.н., доц. Войтко В.В.
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Рецензент: _____
(прізвище та ініціали)

ВНТУ– 2019

Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра програмного забезпечення
Ступінь вищої освіти – магістр
Спеціальність 121 – Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ
Завідувач кафедри ПЗ
Романюк О. Н.
"____"_____20___року

З А В Д А Н Н Я
НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ
СТУДЕНТУ
Кобисі Ігорю Володимировичу

1. Тема роботи: **«Розробка методу і програмних засобів аналізу тональності тексту з використанням нейронних мереж та веб-технологій»**
керівник роботи: к.т.н., доцент кафедри ПЗ Войтко В. В. затверджені наказом вищого навчального закладу від "___"_____20__року №__

2. Строк подання студентом роботи _____

3. Вихідні дані до роботи :
Мова програмування: Pyton
Технологія розробки: Веб-розробка
Браузери (або ОС): Windows, Chrome.

4.Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити): вступ; аналіз стану питання та постановка задачі; розробка модулів додатку; програмна реалізація додатку; тестування розробленого програмного продукту; економічна частина; висновки; перелік посилань; додатки.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень): мета роботи, об'єкт та предмет дослідження; основні задачі дослідження; наукова новизна, практичне значення; архітектура системи; модулі додатку; приклад вигляду розробленої програми; висновки.

6. Консультанти розділів магістерської кваліфікаційної роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видано	завдання прийнято
1–4	к.т.н. Войтко В. В., доцент кафедри ПЗ		
5	к.е.н. Бальзан М. В., доцент кафедри ЕПВМ		

7. Дата видачі завдання_____

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів магістерської кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
	Техніко-економічне обґрунтування доцільності розробки додатку для аналізу тональності тексту		Вик.
	Розробка модулів додатку для автоматизованого аналізу тональності тексту		Вик.
	Програмна реалізація додатку		Вик.
	Тестування роботи додатку		Вик.
	Економічне обґрунтування розробки програмного продукту		Вик.
	Оформлення матеріалів до захисту МКР		Вик.

Студент _____ **Кобися І. В.**

(підпис) (прізвище та ініціали)

Керівник магістерської кваліфікаційної роботи _____ **Войтко В. В.**

(підпис) (прізвище та ініціали)

АНОТАЦІЯ

У магістерській кваліфікаційний роботі розроблено веб додаток для аналізу тональностей тексту з використанням загорткових нейронних мереж, рекурентних нейронних мереж та попередньо навчених моделей нейронних мереж. Програмний додаток призначений для аналізу настроїв у моніторингу соціальних медіа, у моніторингу брендів, у відгуках клієнтів, у службі підтримки, у аналітиці робочої сили та голосу працівників, у аналітиці продукції, у дослідження та аналізі ринку.

Запропоновано метод інтелектуального аналізу даних для визначення тональності тексту, який об'єднує методи словникового аналізу, блочного аналізу та машинного навчання з метою підвищення об'єктивності результатів. Розроблені моделі системи визначення тональності тексту відрізняються від існуючих використанням спеціалізованої нейромережі, призначеної для лінгвістично-емоційного аналізу текстових документів, що підвищує реалістичність результатів ідентифікації текстових файлів.

Створений програмний продукт написаний мовою програмування Python, характеризується зручністю та зрозумілістю інтерфейсу, швидкістю та точністю опрацювання даних, що забезпечує всі вимоги користувача щодо аналізу тональностей тексту.

ABSTRACT

In the master's qualification, a web application was developed for analyzing text tonalities using wrapped neural networks, recurrent neural networks, and pre-trained neural network models. The software application is designed to analyze the sentiment of social media monitoring, brand monitoring, customer feedback, support, workforce and employee voice analytics, product analytics, market research and analysis.

A data mining method is proposed to determine the tone of text that integrates methods of vocabulary analysis, block analysis, and machine learning to increase the objectivity of results. Models of the system of definition of a tone of the text are developed, which differ from the existing ones using specialized neural network, intended for linguistic-emotional analysis of text documents, which increases the realistic results of identification of text files.

The created software product is written in Python programming language, characterized by the convenience and comprehensibility of the interface, the speed and accuracy of data processing, which provides all the requirements of the user to analyze the tone of the text.

ЗМІСТ

ВСТУП

Обґрунтування вибору теми дослідження. Про обробку природної мови сьогодні багато говорять – причому, не тільки в наукових колах, де ця концепція справедливо вважається основним для подальшого розвитку штучного інтелекту, а й представників IT-індустрії.

Серед найбільш цікавих і популярних методів цього широкого наукового напрямку осібно стоїть один, що носить назву sentiment analysis, аналіз тональності текстів. Загальне визначення свідчить, що аналіз тональності текстів – це клас методів контент-аналізу, призначений для автоматичного виявлення в тексті емоційно забарвленої лексики, а також думок (емоційних оцінок) автора з приводу об'єктів, в яких йде мова в тексті. З визначення можна зробити кілька висновків про те, де концепція аналізу тональності тексту могла б знайти застосування і прояснити деякі її деталі[1].

По-перше, аналіз тональності текстів здатний допомогти розібратися в законах, за якими живе природна мова і навчити комп'ютер сприймати його на рівні, наближеному до людського. До недавнього часу машина розуміла тексти на абстрактному рівні - в основному, через лексеми (слова), які для неї мали формою (набір букв) і змістом (значення). Дана концепція пропонує ввести ще одну функцію – так звану лексичну тональність тексту (в найпростішому випадку вона буде визначатися як сума лексичних тональностей кожної окремої лексеми, з яких складається текст).

По-друге, аналіз тональності здатний значно покращити якість перекладів. Відомо, що еталоном машинного перекладу служить результат перекладу тексту людиною - професійним перекладачем. За більше ніж п’ятдесят років розробок в цій області дослідники переконалися в тому, що навчити машину «думати, як перекладач» можна лише взявши до уваги всі ті міркування, якими користується професіонал, переводячи той чи інший текст. При перекладі не обійтися без первинного аналізу тексту та окремих слів - в тому числі, аналізу тональності як такої.

По-третє, метою аналізу тональності тексту може бути якесь думка автора або сам автор. Це - найбільш цікава сфера застосування, оскільки тут бачиться не тільки спосіб делегування машині деяких повноважень вченого (наприклад, філолога, який досліджує твір того чи іншого автора), але і знову спроба наблизити образ мислення комп'ютера до людського. З цієї точки зору аналіз тональності, можливо, є одним з найбільш важливих і перспективних кроків до розвитку штучного інтелекту.

Мета та завдання дослідження. Метою роботи є підвищення рівня реалістичності аналізу та ідентифікації текстових документів за рахунок розробки засобів визначення тональності тексту, що дозволяють автоматизувати процеси обробки даних. Особлива увага в дослідженні буде приділятися використанню нейронних мереж та машинного навчання для розв’язування задачі аналізу тональності заданого тексту. Для досягнення поставленої мети слід розв’язати наступні **задачі**:

- аналіз і порівняння існуючих підходів до визначення тональності тексту;
- аналіз методів дослідження тональностей тексту;
- розробка методу визначення тональності тексту;
- розробка програмного додатку, який буде використовувати створений метод на основі нейронних мереж;
- тестування створеного програмного продукту.

Об’єкт дослідження є процеси аналізу текстової інформації. Предметом дослідження є засоби ідентифікації тональності тексту.

Предметом дослідження є засоби ідентифікації тональності тексту.

Зв’язок роботи з науковими програмами, планами, темами. Робота виконувалася відповідно до плану науково-дослідних робіт кафедри програмного забезпечення.

Методи дослідження:

- методи аналізу тональностей текстів для ідентифікації стилю і тональності текстових файлів;
- теорія нейромереж для розробки метду визначення тональності тексту;

- методи теорії алгоритмів для розробки алгоритмів роботи програми;
- методи програмування для розробки програмного додатку визначення тональності тексту;
- комп'ютерне моделювання для аналізу і перевірки достовірності отриманих теоретичних результатів.

Наукова новизна одержаних результатів:

1. Подальшого розвитку дістав метод інтелектуального аналізу даних для визначення тональності тексту, який відрізняється від існуючих поєднанням підходів словникового аналізу, блочної ідентифікації та машинного навчання, що дозволить підвищити об'єктивність результатів аналізу та забезпечить високу ефективність роботи системи.

2. Подальшого розвитку дістали моделі системи визначення тональності тексту, які, на відміну від існуючих, орієнтовані на розробку та тренування спеціалізованої нейромережі, призначеної для лінгвістично-емоційного аналізу текстових документів, що підвищує реалістичність результатів ідентифікації текстових файлів.

Практична цінність одержаних результатів. Розроблена автоматизована система аналізу тональності тексту. Додаток призначений для автоматизованого аналізу загальнодоступних текстів, визначення ефективності реклами, відношення до обслуговуючого персоналу, визначення якості продукції та наданих послуг через аналіз введеної інформації користувачами у відгуках, що обумовлює затребуваність програмного продукту і перспективність його використання у комерційних цілях.

Зв'язок роботи з науковими програмами, планами, темами. Робота виконувалася відповідно до плану науково-дослідних робіт кафедри програмного забезпечення.

Особистий внесок здобувача В публікації [2] автором розроблено модель роботи системи автоматизованого визначення тональності тексту. В публікації [3] автором проведено аналіз тексту веб-ресурсу. Усі наукові результати, викладені у магістерській кваліфікаційній роботі, отримані автором особисто.

Моделі, алгоритми та програмне забезпечення були розроблені для автоматизованої системи визначення тональності тексту, що реалізована з

використанням мови Python, яка може широко використовуватися для виконання задач машинного навчання та наукових розрахунків.

Розроблений додаток можна використовувати в якості готового продукту.

Достовірність теоретичних положень підтверджена результатами тестування розробленого додатку.

Апробація матеріалів магістерської кваліфікаційної роботи. Результати роботи доповідалися на двох науково-технічних конференціях: на Міжнародній науково-практичній Інтернет-конференції «Електронні інформаційні ресурси: створення, використання, доступ – 2019» та на XLVII Міжнародній науково-технічній конференції ВНТУ – 2018.

Публікації. Результати роботи опубліковані в 2 наукових публікаціях: Розробка засобів аналізу тональностей тексту [2] та створення стайту «Вінниця – туристична перлина Поділля» [3].

Структура та обсяг роботи. Магістерська кваліфікаційна роботи складається зі вступу, п’яти розділів, висновків, списку літератури, що містить 23 найменування, 3 додатків. Робота містить 23 ілюстрації, 5 таблиць.

Робота складається з 5 розділів. У першому розділі проаналізовано сучасний стан питання теми розробки, проведено аналіз підходів до визначення тональності тексту, сформульовано задачі дослідження.

Другий розділ містить розробку методу, моделей та засобів аналізу тональностей текстів.

У третьому розділі розроблено інтерфейс програми, обґрунтовано вибір засобів реалізації системи автоматизованої ідентифікації тональності тексту, проведена програмна реалізація базових модулів системи.

Четвертий розділ показує результати тестування розробленої програми.

У п’ятому розділі розраховуються витрати на розробку програмного забезпечення, експлуатаційні витрати, обсяг роботи, пов'язаної з використанням програмного забезпечення, та економічний ефект від впровадження нового програмного продукту. У висновках наведені основні результати дослідження.

У додатках міститься технічне завдання, лістинг коду основних частин програми та ілюстративний матеріал до захисту магістерської кваліфікаційної роботи.

1 АНАЛІЗ ОСОБЛИВОСТЕЙ ВИЗНАЧЕННЯ ТОНАЛЬНОСТІ ТЕКСТІВ ТА ПОСТАНОВКА ЗАДАЧ РОЗРОБКИ

1.1 Аналіз особливостей визначення тональності текстів

Аналіз тональності тексту (сентимент-аналіз, англ. Sentiment analysis, англ. Opinion mining) – це клас методів аналізу контенту в комп'ютерній лінгвістиці, що призначений для автоматизованого виявлення в текстах емоційно забарвленої лексики і емоційної оцінки автора тексту по відношенню до об'єктів, мова про які йде в тексті [1].

Тональність – це ставлення автора висловлювання до того, про що йде мова в тексті (об'єкту реального світу, події, процесу або їх властивостей), виражене в тексті. Емоційна складова, виражена на рівні лексеми або комунікативного фрагмента, називається лексичною тональністю (або лексичним сентиментом). Тональність всього тексту в цілому можна визначити як функцію (в найпростішому випадку суму) лексичних тональностей складових його одиниць (речень) і правил їх поєднання.

Аналіз тональності текстів – це поле для обробки природних мов (Natural language processing, NLP), яке будує системи, які намагаються ідентифікувати і витягти думки в тексті [2]. Зазвичай, крім ідентифікації думки, ці системи витягують атрибути виразу, наприклад:

– полярність: оратор висловлює позитивну чи негативну думку;

– тема: те, про що говорять;

– власник думки: особа або організація, яка висловлює свою думку.

В даний час аналіз настроїв є цікавим напрямом, що розвивається, оскільки має багато практичних застосувань. Оскільки публічно та приватно доступна інформація в Інтернеті постійно зростає, значна кількість текстів, що висловлюють свої думки, доступна на сайтах рецензування, форумах, блогах і соціальних медіа.

За допомогою систем аналізу настроїв ця неструктурована інформація може автоматично трансформуватися у структуровані дані громадської думки про

продукти, послуги, бренди, політику або будь-яку тему, щодо якої люди можуть висловлювати свої думки. Ці дані можуть бути дуже корисними для комерційних додатків, таких як маркетинговий аналіз, зв'язки з громадськістю, огляди продуктів, підрахунок чистого промоутера, відгуки про продукцію та обслуговування клієнтів.

Текстову інформацію можна розділити на два основні типи: факти та думки. Факти є об'єктивними виразами про щось. Думки – це, як правило, суб'єктивні вирази, що описують почуття, оцінки та відношення людей до події чи теми.

Аналіз сентиментів, як і багато інших задач NLP, може бути змодельований як проблема класифікації, де необхідно вирішити дві підпроблеми:

– класифікація речення як суб'єктивна або об'єктивна, відома як класифікація суб'єктності;

– класифікація речення як вираження позитивної, негативної або нейтральної думки, відома як класифікація полярності.

На нашу думку, об'єкт, про який говорить текст, може бути об'єктом, його складовими, його аспектами, його атрибутами або його особливостями. Це також може бути продукт, послуга, особа, організація, подія або тема. Наприклад, у реченні: «Термін служби акумулятора цієї камери надто короткий» – висловлюється негативна думка про особливість (час автономної роботи) суб'єкта (камери).

Існує два види думок: пряма і порівняльна. Думка першого типу може бути визначена формально: простою думкою називається кортеж з п'яти елементів (entity, feature, sentiment value, holder, time). В цьому визначенні автор (holder) висловив думку про аспект (feature) об’єкту entity в певний момент часу (time). Зазвичай виділяють два види емоцій (sentiment value): позитивні та негативні, тобто класифікація виконується за двома класами. Іноді додається третій – нейтральна думка. Прямі думки дають висновок про суб'єкт безпосередньо, наприклад: «Якість зображення камери А погана». Ця пряма думка висловлює негативну оцінку камери А.

Другий тип думок – порівняння – можна розділити на три види:

- порівняння аспектів об'єктів на користь одного (non–equal gradable);

- прирівнювання аспектів різних об'єктів (equative);

- перевага одного об'єкта над іншими (superlative).

Порівняння першого типу мають вигляд «аспект об'єкта 1 перевершує в чомусь аспект об'єкта 2», наприклад: «Ціни в інтернет-магазині «А» нижче, ніж в інтернет-магазині «В».

Другий тип виражає схожість аспектів різних об'єктів, наприклад: «Ціни в інтернет-магазинах «А» та «Б» майже однакові».

Прикладом третього типу може слугувати речення «В конкурсі на кращий магазин місяця магазин «А» перемагає магазин «Б».

Думка другого типу визначається як кортеж (Obj1, Obj2, A, holder, time). В даному кортежі Obj1 и Obj2 – множини порівнюваних за аспектом А об'єктів, які автор (holder) порівнює у момент часу time. На відміну від кортежу, який визначає думку першого типу, кортеж думки другого типу не містить прямої оцінки емоцій автора.

В аналізі тональності тексту часто зустрічається термін, пов'язаний з поняттям думки – суб'єктивність. Визначення об'єктивного і суб'єктивного речень наступне:

- об'єктивне речення відображає фактичну інформацію про що-небудь, тоді як суб'єктивне речення виражає чиїсь особисті почуття і припущення;

- об’єктивні речення зазвичай не мають емоційного забарвлення, тому аналіз тексту на наявність суб'єктивної інформації часто є підзадачею визначення полярності тексту.

У комп'ютерній лінгвістиці текст природною мовою вважається неструктурованою інформацією. У завданнях, об'єднаних терміном «аналіз емоційного забарвлення тексту», визначається те, яким чином з тексту природною мовою витягується, аналізується і структурується інформація.

Отже, аналіз тональності тексту зазвичай включає в себе наступні основні завдання:

- визначення наявності емоційного забарвлення;

- визначення полярності тексту;

- вилучення аспектів з емоційно забарвленого тексту.

Задача визначення полярності тексту формулюється наступним чином: «визначити, яке емоційне забарвлення тексту, позитивне чи негативне?» Визначення полярності тексту зазвичай розглядається на декількох рівнях:

- на рівні документу. Основною задачею на цьому рівні є класифікація, чи повністю весь документ є відображенням позитивної чи негативної думки. Наприклад, для певної рецензії на товар, система автоматичного аналізу тональності тексту визначає, чи висловлює ця рецензія позивний настрій в цілому. Дана задача носить назву «класифікація полярності настроїв на рівні документу». Такий рівень деталізації передбачає, що кожен документ висловлює думку як єдину сутність. Тому він не може застосовуватись для документів, об'єктами яких є декілька сутностей. Однак в рамках данної задачі, зазвичай документи мають одне чітке емоційне забарвлення, бо коментарі до новин зазвичай є досить короткими емоційними текстами; [4]

- на рівні речення. На цьому рівні об'єктом дослідження є окреме речення. Проводиться аналіз чи висловлює певне речення в цілому позитивну чи негативну думку. Аналіз на даному рівні близько пов'язаний з так званою «класифікацією суб'єктивності» (subjectivity classification), яка розрізняє речення (так звані об'єктивні речення) що висловлюють фактичну інформацію від речень, що висловлюють думки та погляди;

- на рівні сутності та аспекту. Обидва попередніх рівня не включають аналіз того, що саме сподобалося чи не сподобалося власнику думки. Аспектний рівень дозволяє виконати більш детальний аналіз. Замість того, щоб аналізувати лінгвістичні конструкції, аспектний рівень аналізує саму думку. Зазвичай, аналіз думки без аналізу її об'єкту має обмежене використання. Окрім того, визнання важливості аналізу об'єкту думки допомагає глибше зрозуміти проблему аналізу тональності. Наприклад, речення «Не зважаючи на поганий сервіс, мені все одно сподобався цей ресторан» має позитивний сентимент, але ми не можемо стверджувати, що воно є повністю позитивним. Власне, даний сентимент можна

вважати позитивним лише в тому випадку, якщо в якості об’єкту обрано «ресторан». Якщо ж в якості об’єкту обрано «сервіс», то думка є повністю негативною. В багатьох дослідженнях об’єкти думки описуються сутностями та їх аспектами. Тобто, метою аналізу такого рівня є виявлення сентименту об’єкту та його властивостей. На такому рівні аналізу, можна отримати структурований підсумок думок щодо не тільки самого об’єкту, але і його властивостей, що перетворить неструктурований текст в структурований масив даних що може бути використаний для будь-яких типів аналізу.

Взагалі, аналіз на рівні документу та на рівні речення є досить складними, а аналіз на рівні аспектів є неймовірно важкою задачею. Найбільш важливими індикаторами сентименту є «слова емоційного забарвлення» (sentiment words). Ці слова зазвичай використовуються для висловлення думки, позитивної чи негативної. Наприклад, «добре», «чудово», «неймовірно» – слова для висловлення позитивної думки, в той час як «погано», «жахливо», «сумно» – слова для висловлення негативної думки. Окрім безпосередньо слів, існують також фрази та ідіоми. Слова та вирази є інструментарієм аналізу тональності з очевидних причин. Набір таких слів та виразів називається «лексичним словником». Незважаючи на те, що слова та вирази для вираження емоційного забарвлення ж дуже важливими при аналізі тональності, просто використання їх не є досить ефективним. Проблема є комплексною та набагато складнішою.

Проаналізуємо основні проблеми, з якими можна стикнутися при виконанні аналізу тональності за словником:

– позитивне чи негативне слово може приймати протилежний відтінок при використанні в іншій предметній області. Так, наприклад, «У цього фільму передбачуваний сюжет» є негативною характеристикою, а «У цього коду передбачувана поведінка» є позитивною;

– наявність неологізмів та помилок у словах, а це досить поширене у інтернеткультурі явище, у багатьох випадках робить використання словникових методів недоцільним; [4]

– речення, що містить в собі слово емоційного забарвлення, може мати нейтральний сентимент. Цей феномен виникає зазвичай в декількох типах речень. Питальні речення та умовні речення є двома найважливішими типами, наприклад «Чи не могли в Ви порадити, яка з фотокамер Sony є найкращою?», та «Якщо я знайду дійсно хорошу камеру в цьому магазині, я її обов'язково куплю.» В обох цих реченнях є слова, що виражають позитивний настрій («найкраща», «хороша»), але жодне з цих речень не висловлює позитивну чи негативну думку щодо певної камери;

– речення, що містять сарказм з наявністю чи відсутністю слів емоційного забарвлення є дуже складними для аналізу, наприклад «Який чудовий телефон! Перестав працювати вже за два дні». Сарказм досить досить часто з'являється в політичних дебатах чи дискусіях, тому аналізувати політичні думки, як правило, досить важко. Ця проблема останнім часом має багато уваги, і навіть з'являються деякі практичні результати [5].

Всі описані вище проблеми є досить серйозними труднощами для виконання аналізу тональності, що базується на лексичному словнику. Тому, методи, що використовують лексичний словник, як правило, застосовуються рідше, ніж альтернативні методи.

1.2 Визначення типів аналізу тональностей

З ростом популярності і доступності громадської думки на багатьох ресурсах, таких як сайти відгуків та рецензій, особисті блоги, соціальні мережі, тощо, виникають нові можливості і водночас нові проблеми, оскільки користувачі можуть та активно використовують інформаційні технології, щоб шукати і розуміти думки інших. Раптове зростання такої діяльності в галузі емоційно-смислового аналізу, яка має справу з обчислювальною обробкою людських думок, тональності, емоцій і суб'єктивності в тексті [1] відбулося саме в час бурхливого розвитку та зростання інтересу до нових систем, які безпосередньо стосуються думок.

Останніми роками проведені дослідження емоційного забарвлення відгуків про фільми, ресторани, готелі, техніку, політичні події, тощо[4]. В Україні також активно розвивається ця галузь: проводяться дослідження відгуків українською мовою про заклади харчування, ведеться робота над емоційно-смисловим аналізатором української мови [4], розпочато створення сентимент-анотованого корпусу і на його основі генерацію тонального словника шляхом залучення додатково словників синонімів та антонімів [4].

Емоційно-смисловий аналіз або аналіз тональності – це галузь науки, яка займається аналізом людських думок, емоцій, відношення стосовно певних подій, продукції, послуг, організацій, тощо [1]. Під час проведення дослідження спершу необхідно визначити поняття емоційно-смислового аналізу або проблеми аналізу тональності тексту. З дослідницької точки зору, ці поняття дають нам постановку задачі і дозволяють нам бачити весь набір взаємозв'язаних підзадач, які становлять суть проблеми емоційно смислового аналізу [1]. І це вказує на те, що якщо ми не можемо структурувати проблему, то ми, ймовірно, не розуміємо її. Завдання цих визначень полягає, таким чином, у виділенні структури зі складного неструктурованого тексту природною мовою. Вони також служать в якості загальної основи для уніфікації різних існуючих наукових напрямків, і дають можливість дослідникам створювати точніші і більш надійні методи рішень за рахунок використання взаємозв'язків підзадач [1]. З точки зору практичного застосування, ці визначення дають змогу дослідникам бачити те, що потрібно для підзадач, які будуть вирішуватися на практиці, як вони пов'язані між собою, і який результат потрібно отримати. На відміну від фактичної інформації, думки і тональність мають важливу рису – суб'єктивність [3]. Таким чином, необхідно опрацювати думки різних людей, а не тільки погляд однієї людини, тому що така думка відображає тільки суб'єктивний погляд цієї однієї людини, якого, як правило, не достатньо для його застосування в дослідженні. Через поширення великої кількості висловлених думок в Інтернеті, з’являється потреба у доступному емоційно-смисловому аналізаторі [4]. Для цього необхідно сентимент-анотований корпус та згенерований на його основі тональний словник.

Завдяки тональності окремих слів можна визначити тональність всього повідомлення загалом. Емоційне забарвлення може бути позначене як позитивне, негативне чи нейтральне [3]. Наприклад, слова чудовий, щастя, доброта можуть бути позначені як позитивні, а от такі слова як поганий, потворний, сум позначаються як негативні. Тональність – це засіб для визначення, класифікації та вилучення емоційно смислової лексики з речення або більших частин тексту [2].

В сучасних системах автоматичного визначення емоційної оцінки тексту найчастіше використовується одномірний емотивний простір: позитив чи негатив (добре або погано). Однак відомі успішні випадки використання і багатовимірних просторів.

Основним завданням в аналізі тональності є класифікація полярності тексту, тобто визначення, чи є виражена в тексті думка позитивною, негативною або нейтральною. Більш розгорнута класифікація тональності виражається, наприклад, такими емоційними станами, як «злий», «сумний» і «щасливий».

Полярність документа можна визначати за бінарною шкалою. У цьому випадку для визначення полярності документа використовується два класи оцінок: позитивна чи негативна.

Одним з недоліків цього підходу є те, що емоційну складову документа не завжди можна однозначно визначити, тобто документ може містити як ознаки позитивної оцінки, так і негативної ознаки. Ранні роботи в цій галузі включають в себе праці Терні і Панга, які застосовують різні методи розпізнавання полярності оглядів товару і відгуків про фільми відповідно. Це приклад роботи на рівні документа.

Можна класифікувати полярність документа за багатосмуговою шкалою, що було розроблена Пангом і Снайдером (серед інших) [2]. Ними було розширене основне завдання класифікації кіновідгуків від оцінки «позитивний або негативний» в бік прогнозування рейтингу за 3-и або 4-и бальною шкалою. У той же час Снайдер провів поглиблений аналіз оглядів ресторанів, пророкуючи рейтинги їх різних властивостей, таких як їжа і атмосфера (за 5-бальною шкалою).

Іншим методом визначення тональності є використання систем оцінювання, в яких словами, зазвичай пов'язаними з негативними, нейтральними або позитивними тональностями, ставляться відповідно числа за шкалою від -10 до 10 (від найнегативнішого до найпозитивнішого). Спочатку фрагмент неструктурованого тексту досліджується з допомогою інструментів та алгоритмів обробки природної мови, а потім виділені з цього тексту об'єкти та терміни аналізуються з метою розуміння значення цих слів.

Інший дослідницький напрямок – це ідентифікація суб'єктивності / об'єктивності. Це завдання зазвичай визначається як віднесення даного тексту до одного з двох класів – суб'єктивний чи об'єктивний. Ця проблема іноді може бути складнішою, ніж класифікація полярності: суб'єктивність слів і фраз може залежати від контексту, а об'єктивний документ може містити в собі суб'єктивні пропозиції (наприклад, новинна стаття, цитує думки людей). Більш того, як згадував Су[4], результати більшою мірою залежать від визначення суб'єктивності, що вживається в рамках анотації текстів. В будь-якому випадку, Панг показав, що видалення об'єктивних пропозицій з документа перед класифікацією полярності допомогло підвищити точність результатів.

Модель більш докладного аналізу називається аналізом на основі функції/ аспекту. Ця модель посилається на ухвалу думок або настроїв, виражених різними функціями або аспектами сутностей, наприклад, у стільникового телефону, цифрової камери або банку. Властивість/аспект – це атрибут або компонент сутності, що досліджується на тональність, наприклад, екран мобільного телефону або ж якість зйомки камери. Ця проблема вимагає вирішення ряду завдань, наприклад, ідентифікація актуальних сутностей, витяг їх функцій аспектів та визначення, є думка, висловлена по кожній функції/аспекту, позитивною, негативною або нейтральною. Більш докладні дискусії на цей рахунок можуть бути знайдені в довіднику з NLP, у розділі «Аналіз тональності та суб'єктивності» [5].

Існує багато видів аналізу настроїв, а інструменти аналізу тональностей варіюються від систем, які зосереджуються на полярності (позитивні, негативні,

нейтральні) до систем, які виявляють почуття і емоції (сердиті, щасливі, сумні тощо) або визначають наміри (наприклад, зацікавлені чи не зацікавлений).

Інколи можна бути більш точними щодо рівня полярності думки, тому замість того, щоб просто говорити про позитивні, нейтральні або негативні думки, можна розглянути такі категорії:

- дуже позитивно;
- позитивно;
- нейтрально;
- негативно;
- дуже негативно.

Це зазвичай називають дрібнозернистим аналізом. Це може бути, наприклад, нанесено на 5-зірковий рейтинг в огляді, наприклад: дуже позитивний = 5 зірок і дуже негативний = 1 зірка.

Деякі системи також надають різні відтінки полярності, визначаючи, чи позитивні або негативні настрої асоціюються з певним почуттям, таким як гнів, смуток чи турбота (тобто негативні почуття) або щастя, любов чи ентузіазм (тобто позитивні почуття).

Виявлення емоцій спрямоване на виявлення таких емоцій, як щастя, розчарування, гнів, смуток тощо. Значна кількість систем виявлення емоцій вдаються до лексиконів (тобто списків слів і емоцій, які вони передають) або складних алгоритмів машинного навчання.

Одним з недоліків вживання лексиконів є те, що спосіб, у який люди висловлюють свої емоції, сильно змінюється, так само як і лексичні предмети, які вони використовують. Деякі слова, які, як правило, виражають гнів, як лайно або вбивають (наприклад, у вашому продукті це лайно, або ваша клієнтська підтримка вбиває мене), можуть також виражати щастя (наприклад, у таких текстах, як Це лайно, або Ви його вбиваєте).

Як правило, під час аналізу настроїв у суб'єктах, наприклад, на продуктах, вас може зацікавити не тільки те, що люди говорять з позитивною, нейтральною або негативною полярністю про продукт, а й які конкретні аспекти або

особливості продукту, про який люди говорять . Ось про що йдеться в аспект-аналізі. У нашому попередньому прикладі: "Термін служби акумулятора цієї камери надто короткий".

Пропозиція висловлює негативну думку про камеру, а точніше, про час автономної роботи, що є особливою характеристикою камери.

Аналіз наміру в основному виявляє те, що люди хочуть робити з текстом, а не те, що люди говорять з цим текстом. Подивіться на наступні приклади.

Приклад перший: “Ваша підтримка клієнтів є катастрофою. Я тримаюся 20 хвилин”.

Приклад другий: “Я хотів би знати, як замінити картридж”.

Приклад третій: "Чи можете ви допомогти мені заповнити цю форму?"

Людина не має проблем з виявленням скарги в першому тексті, питанням у другому тексті, а запит в третьому тексті. Тим не менш, машини можуть мати деякі проблеми, щоб визначити їх. Іноді передбачувана дія може бути виведена з тексту, але іноді виведення цього вимагає певного контекстуального знання, для якого штучному інтелекту потрібна додаткова інформація.

Багатомовний аналіз настроїв може бути складним завданням. Як правило, необхідна велика кількість попередньої обробки, і попередня обробка використовує ряд ресурсів. Більшість цих ресурсів доступні в Інтернеті (наприклад, лексикони сентиментів), але потрібно створити багато інших (наприклад, перекладені корпуси або алгоритми виявлення шуму). Використання доступних ресурсів вимагає багато досвіду кодування і може зайняти багато часу для реалізації.

1.3 Аналіз існуючих методів аналізу тональностей

Існує багато методів і алгоритмів для реалізації систем аналізу настроїв, які можна класифікувати як:

– системи на основі правил, які виконують аналіз настроїв на основі набору ручних правил;

– автоматичні системи, які спираються на методи машинного навчання, щоб дізнатися з даних;

– гібридні системи, які поєднують обидві правила і автоматичні підходи.

Як правило, підходи, що базуються на правилах, визначають набір правил у певній скриптовій мові, яка ідентифікує суб'єктність, полярність або предмет думки.

Правила можуть використовувати різні входи, такі як:

– класичні методи НЛП, такі як витіснення, маркування, частина мічення мовлення і розбору;

– інші ресурси, такі як лексикони (тобто списки слів і виразів).

Основним прикладом реалізації на основі правил буде наступне:

Визначте два списки поляризованих слів (наприклад, негативні слова, такі як погані, найгірші, потворні тощо, і позитивні слова, такі як хороші, найкращі, красиві тощо).

Дано текст.

Підрахуйте кількість позитивних слів, які з'являються в тексті.

Підрахуйте кількість негативних слів, які з'являються в тексті.

Якщо кількість позитивних виступів слів більше, ніж кількість негативних слів, то повернення позитивних настроїв, навпаки, повернення негативних настроїв. В іншому випадку поверніть нейтраль.

Ця система дуже наївна, оскільки не враховує, як слова поєднуються в послідовності. Можна зробити більш просунуту обробку, але ці системи швидко стають дуже складними. Їх дуже важко підтримувати, оскільки можуть знадобитися нові правила, щоб додати підтримку для нових виразів і словника. Крім того, додавання нових правил може мати небажані наслідки в результаті взаємодії з попередніми правилами. Як результат, ці системи вимагають важливих інвестицій в ручне налаштування та підтримку правил.

Автоматичні методи, всупереч системам на основі правил, не покладаються на правила, створені вручну, а на техніку машинного навчання. Завдання аналізу настроїв зазвичай моделюється як проблема класифікації, коли класифікатор

подається текстом і повертає відповідну категорію, наприклад, позитивний, негативний або нейтральний (у випадку, якщо проводиться аналіз полярності).

Перший крок у класифікаторі тексту машинного навчання полягає в перетворенні тексту в числове представлення, зазвичай вектор. Зазвичай кожен компонент вектора являє частоту слова або виразу в попередньо визначеному словнику (наприклад, лексикон поляризованих слів). Цей процес відомий як екстракція ознак або векторизація тексту, а класичний підхід був мішкою слів або мішкою з їх частотою.

Зовсім недавно були застосовані нові методи вилучення ознак на основі вбудовування слів (також відомих як вектори слів). Таке уявлення дає можливість для слів з подібним значенням мати подібне уявлення, що може поліпшити продуктивність класифікаторів.

Крок класифікації, як правило, включає статистичну модель, таку як Naive Bayes, логістична регресія, машини підтримки векторних чи нейронних мереж.

1.4 Визначення метрик аналізу тональностей текстів

Існує багато способів, за допомогою яких можна отримати показники ефективності для оцінки класифікатора і зрозуміти, наскільки точна модель аналізу настроїв. Один з найбільш часто використовуваних називається перехресною перевіркою.

Що робить перехресна перевірка - це поділ тренувальних даних на певну кількість складних даних (з 75% навчальних даних) і таку ж кількість тестових даних (з 25% навчальних даних), використання тренувальних даних для навчання Класифікатора, і перевірити його на тестування складок, щоб отримати показники продуктивності. Процес повторюється кілька разів і обчислюється середня для кожної метрики.

Якщо тестовий набір завжди один і той же, це означає, що набір перенасичений, що означає, що можна скоригувати свій аналіз для даного набору даних так, що не можна буде проаналізувати інший набір. Перехресна перевірка

допомагає запобігти цьому[3]. Чим більше даних, тим більше різних наборів можна використовувати.

Точність, продуктивність та час відклику є стандартними показниками, які використовуються для оцінки продуктивності класифікатора.

Точність вимірює, скільки текстів було передбачено правильно як приналежність до даної категорії з усіх текстів, які були передбачені (правильно і неправильно) як належать до категорії.

Нагадаємо про міри, скільки текстів було передбачено правильно як приналежність до даної категорії з усіх текстів, які повинні були передбачати, що належать до категорії. Ми також знаємо, що чим більше даних ми будемо годувати нашими класифікаторами, тим краще буде згадувати.

Найчастіше для вимірювання продуктивності використовують точність і час відклику, оскільки тільки точність не говорить про те, наскільки хорошим чи поганим є класифікатор.

Для складних завдань, таких як аналіз настроїв, точність і час відклику, напевно, будуть низькими. Після подання класифікатору більше даних продуктивність буде покращена. Проте, як ми побачимо нижче, оскільки анотовані дані навряд чи будуть точними, є ймовірність того, що рівні точності не стануть надто високими. Однак, якщо подавати класифікатору послідовно розмічені даних, результати будуть настільки ж хорошими, як результати можуть бути для будь-якої іншої проблеми класифікації.

Коли мова йде про угоду між анотаторами (тобто домовленість людей щодо даного завдання анотації), однією з найбільш часто використовуваних показників є криппендорфська альфа. На думку Сайфа та ін., найкраща угода між анотаторами для аналізу настроїв у Twitter досягає значення 0.655 для Alpha у Krippendorff. Це означає, що є багато угод (оскільки альфа більше нуля), але ми вважаємо, що це ще далеко від великого (наприклад: близько 0,8, що є мінімальним порогом надійності, яке використовують соціологи для того, щоб сказати, що дані є надійними).

1.5 Аналіз сфер використання тональностей текстів

Під базовими сферами використання тональностей текстів розуміємо аналіз настроїв у моніторингу ресурсів соціальних медіа, моніторинг брендів, аналіз настроїв у відгуках клієнтів, аналіз настроїв у службі підтримки клієнтів, аналіз настроїв у аналітиці робочої сили і голосу працівника та визначення тональностей у аналітиці продукту, аналіз настроїв у дослідженні та аналізі ринку.

1.5.1 Аналіз настроїв у моніторингу ресурсів соціальних медіа

У фатальний вечір 9 квітня 2017 року United Airlines примусово вивезла пасажира з перенавантаженого рейсу. Інцидент з кошмаром був знятий іншими пасажирами на їхніх смартфонах і негайно розміщений [6]. Одне з таких відео, розміщене на Facebook, було більше 87 тисяч разів і було переглянуто 6,8 мільйона разів у 6 вечора в понеділок, лише через 24 години.

Фіаско жахливо звеличувалося завдяки репрезентативній реакції компанії. У понеділок після обіду вони надіслали заяву від генерального директора про вибачення за те, що їм доведеться повторно розмістити клієнтів.

Це саме той тип PR-катастрофи, без якого ми всі хотіли б щасливо працювати. Це також чудовий приклад того, чому ми піклуємося не тільки про те, що люди говорять про наш бренд, але й про те, як вони це говорять. Більше згадувань не дорівнює позитивним згадкам.

У нинішній час, бренди всіх форм і розмірів мають значущі взаємодії з клієнтами, провідниками і навіть конкуренцією в соціальних мережах, таких як Facebook, Twitter і Instagram. Більшість маркетингових департаментів вже налаштовані на онлайн-згадки щодо обсягу - вони оцінюють більше балачки, оскільки більше знань про бренд. Сьогодні, однак, ми можемо зробити ще глибше. Використовуючи аналіз настроїв у соціальних мережах, ми можемо отримати неймовірну інформацію про якість розмови, що відбувається навколо бренду.

Як можна використовувати аналіз настроїв:

– проаналізуйте твіти та / або повідомлення на Facebook протягом періоду часу, щоб побачити почуття певної аудиторії;

– запустіть аналіз настроїв у всіх згаданих соціальних мережах до вашого бренду та автоматично класифікуйте їх за терміном;

– автоматично направляйте повідомлення про соціальні медіа для членів команди, які найкраще відповідатимуть;

– автоматизуйте будь-який або всі ці процеси;

– використовуйте аналітику, щоб отримати глибоке розуміння того, що відбувається на ваших каналах соціальних медіа.

Переваги використання аналізу тональностей тексту для моніторингу соціальних мережах:

– визначення пріоритетності дій. Що більш терміново: димлячий клієнт або тонке "спасибі!" Очевидно, споживач. Аналіз настроїв дозволяє легко фільтрувати непрочитані згадки за допомогою позитивності та негативу, показуючи вам, які палаючі вогні покласти на список "негайно погасити" і які повільні тлеючі можуть трохи почекати;

– відстежування тенденції з плином часу;

– визначення найбільш прийнятного моменту для запуску нового продукту чи відстеження зменшення обсягу продаж;

– відстеження соціальних мереж конкурентів, що може допомогти, наприклад, не повторювати помилок конкурентів (якщо якийсь новий продукт покупцям не сподобався або рекламна компанія, наприклад, виявилася невдалою, цей досвід можна використати):

– використовуйте аналітику, щоб отримати глибоке розуміння того, що відбувається на ваших каналах соціальних медіа.

1.5.2 Аналіз настроїв у моніторингу брендів

Не тільки бренди мають багату інформацію, доступну в соціальних мережах, але вони також можуть виглядати більш широко в Інтернеті, щоб побачити, як люди говорять про них в Інтернеті. Замість того, щоб зосередитися

на певних соціальних медіа-платформах, таких як Facebook і Twitter, ми можемо націлювати згадки в таких місцях, як новини, блоги та форуми - знову, не тільки обсяг згадувань, а й якість цих згадок.

У прикладі United Airlines, наприклад, почалася розгорання ситуації на платформах соціальних медіа декількох пасажирів. Протягом декількох годин він був підхоплений новинними сайтами і поширився як лісова пожежа по США. Потім новини поширилися в Китай і В'єтнам, оскільки пасажир, як повідомлялося, був американцем китайсько-в'єтнамського походження, і люди звинувачували винних у расовому профілюванні. У Китаї інцидент став трендовою темою номер один на сайті Weibo, мікроблогів з майже 500 мільйонами користувачів[7].

І знову ж таки, все це відбувається протягом кількох годин і днів, коли стався інцидент.

Як можна використовувати аналіз настроїв:

– проаналізуйте новини, публікації в блогах, обговорення на форумі та інші тексти в Інтернеті протягом певного періоду часу, щоб побачити почуття певної аудиторії;

– автоматично класифікуйте терміновість всіх онлайнових згадок вашого бренда через аналіз настроїв;

– автоматично сповіщати призначених членів групи про онлайн-згадки, які стосуються їхньої сфери діяльності;

– автоматизуйте будь-який або всі ці процеси;

– краще зрозумійте присутність бренду в Інтернеті, отримуючи всілякі цікаві ідеї та аналітику для подальшого розвинення бренду, беручі за основу справжні клієнтські відгуки.

Аналіз спостережень корисний для моніторингу бренду, оскільки він допомагає виконувати наступне:

– розуміння, як з часом розвивається репутація марки;

– дослідження конкурентів і розуміння, як з часом змінюється їхня репутація;

– визначення потенційні кризи в сфері зв'язків з громадськістю і знання, що необхідно вжити негайних заходів;

– фокус на певному момент часу. Знову ж таки, можливо, користувач хочете подивитися лише згадки про пресу в день подачі заявки на IPO, або запуск нового продукту. Аналіз настроїв дозволяє це зробити.

Приклад: Expedia Canada. Біля Різдва, Expedia Canada провела класичну маркетингову кампанію "Зимова втеча". Все було добре, за винятком вибору скрипки як фонової музики. Зрозуміло, що люди потрапили до соціальних медіа, блогів і форумів. Expedia помітив і видалив оголошення. Потім вони створили серію відеозаписів: один показав, що оригінальний актор розбив скрипку, а в іншому – запросив справжнього послідовника, який скаржився в Twitter, щоб він прийшов і зірвав скрипку.

Хоча їх оригінальний продукт був далеко не бездоганним, врешті решт вони змогли викупити себе, включивши реальні відгуки клієнтів до продовжених ітерацій.

Використовуючи аналіз настроїв (і машинне навчання), можна автоматично контролювати всі розмови та відгуки навколо свого бренду і виявляти цей тип потенційно-вибухового сценарію, поки у вас ще є час для його розрядки[8].

1.5.3 Аналіз настроїв у відгуках клієнтів

Соціальні медіа та моніторинг бренду пропонують нам негайну, нефільтровану, безцінну інформацію про настрої клієнтів. У паралельному ключі запускаються дві інші команди розуміння - опитування та взаємодія з клієнтською підтримкою. Команди часто дивляться на свій чистий показник промоутера (NPS), але ми також можемо застосувати цей аналіз до будь-якого типу обстеження або каналу зв'язку, що дає можливість отримати текстові відгуки клієнтів.

Опитування NPS задають кілька простих запитань, а саме: чи рекомендували б ви цю компанію, продукт і / або послугу своєму другові або

члену сім'ї? і чому? - і використовувати це для ідентифікації клієнтів як промоутерів, пасивів або недоброзичливців.

Мета полягає в тому, щоб визначити загальний досвід клієнтів, і знайти способи підняти всіх клієнтів на рівень «промоутер», де вони теоретично купуватимуть більше, залишатимуться довше і направлятимуть інших клієнтів.

Дані чисельних опитувань легко агрегуються та оцінюються, але ми хочемо, щоб таку ж легкість відповідали і на питання «чому». Регулярний показник NPS просто дає вам номер, без додаткового контексту того, про що йде мова і чому оцінка приземлилася там. Аналіз сприйняття робить цей крок подальшим.

Як можна використовувати аналіз настроїв:

– аналіз агрегованих NPS або інших відповідей на дослідження;

– проаналізуйте агреговані взаємодії з клієнтською підтримкою;

– відстежуйте настрої клієнтів щодо конкретних аспектів бізнесу з плином часу. Це додає глибину, щоб пояснити, чому загальна оцінка NPS може змінитися, або якщо окремі аспекти змінилися незалежно;

– орієнтуйте людей на поліпшення їхнього обслуговування. Автоматизуючи аналіз настроїв на вхідні опитування, ви можете бути попереджені клієнтам, які відчувають негативний вплив на ваш продукт або послугу, і можуть конкретно займатися ними;

– визначте, чи конкретні клієнтські сегменти відчувають себе більш чітко про вашу компанію. Деякі демографічні показники, інтереси, персонажі та ін.

Аналіз настроїв корисний для розуміння Голосу Клієнта (VoC), оскільки він допомагає виконувати наступні дії:

– використовуйте результати аналізу настроїв для розробки більш поінформованих питань, щоб задати питання про майбутні дослідження.

– зрозумійте нюанси клієнтського досвіду з часом, а також чому і як відбуваються зміни;

– посилюйте свої внутрішні команди, надаючи їм більш глибокий погляд на досвід клієнтів, на сегменти та на конкретні аспекти бізнесу;

– швидше реагуйте на сигнали та переходи від клієнтів.

Приклад: Проект голосів McKinsey.

McKinsey & Company - міжнародна консалтингова компанія, що спеціалізується на вирішенні завдань, пов'язаних зі стратегічним управлінням. McKinsey як консультант співпрацює з найбільшими світовими компаніями, державними установами і некомерційними організаціями.

У Бразилії федеральні державні витрати зросли на 156% з 2007 по 2015 роки, в той час як задоволеність населення державними послугами постійно зменшувалася. Незадоволений цим контрпродуктивним прогресом, Департамент міського планування набрав McKinsey, щоб допомогти їм працювати над низкою нових проектів, які б зосередилися в першу чергу на досвіді користувачів або подорожах громадян під час надання послуг.

Цей стиль управління, орієнтований на громадян, призвів до того, що ми називаємо Розумні міста.

McKinsey розробила інструмент, який називається City Voices, який проводить опитування громадян (клієнтів) у більш ніж 150 різних показниках, а потім проводить аналіз настроїв, щоб допомогти керівникам зрозуміти, як живуть складові і що їм потрібно, щоб краще інформувати державну політику. Використовуючи цей інструмент, бразильський уряд зміг розв'язати нагальні потреби - наприклад, безпечнішу автобусну систему - і в першу чергу поліпшити їх.

Якщо навіть цілі міста та країни, відомі своєю бюрократією та повільними темпами, включають подорожі клієнтів та аналіз настроїв у свої процеси прийняття рішень, то інноваційні компанії краще будуть далеко попереду.

1.5.4 Аналіз настроїв у службі підтримки клієнтів

Відомо, що тренування: зоряний досвід клієнтів = більш ймовірні клієнти, що повертаються. Особливо в останні роки було багато розмов (по праву) навколо досвіду клієнтів та подорожей клієнтів. Провідні компанії почали усвідомлювати, що найчастіше те, як вони поставляють, є такими ж (якщо не більш) важливими,

як те, що вони забезпечують. Сьогодні, як ніколи раніше, клієнти очікують, що їхній досвід роботи з компаніями буде негайним, інтуїтивним, особистим та безпроблемним. Фактично, дослідження показують, що 25% клієнтів перейдуть на конкурента після одного негативного взаємодії.

Провідні компанії почали усвідомлювати, що найчастіше те, як вони поставляють, є такими ж (якщо не більш) важливими, як те, що вони забезпечують.

Як можна використовувати аналіз настроїв:

– автоматизація систем для виконання аналізу настроїв на всі вхідні запити щодо підтримки користувачів;

– швидко виявляйте розчарованих клієнтів;

– маршрут запитів до конкретних членів команди найкраще підходить для відповіді;

– використовуйте аналітику, щоб отримати глибоке розуміння того, що відбувається в службі підтримки клієнтів.

Аналіз настроїв корисний у підтримці клієнтів, оскільки допомагає виконувати наступне:

– визначте пріоритетність порядку реагування на квитки, переконавшись, що в першу чергу вирішимо найбільш нагальні потреби;

– підвищуйте ефективність, автоматично призначаючи квитки певній категорії або члена команди.

Приклад: Аналіз взаємодії з клієнтською підтримкою в Twitter.

Було проведено аналіз того, як чотири найбільших американських перевізників телефону (AT&T, Verizon, Sprint, та T-Mobile) обробляли взаємодію підтримки клієнтів у Twitter.

Було завантажено десятки тисяч твітів, де були згадані ці компанії (по імені або за допомогою вказівки на користувача), і провели їх через модель MonkeyLearn, щоб класифікувати кожен твіт як позитивний, нейтральний або негативний. Потім ми використали наш новий Insight Extractor, який читає весь текст як одну одиницю, після чого витягує найбільш релевантні ключові слова і

повертає найбільш відповідні речення, включаючи кожне окреме ключове слово, що було виявлено.

Ось деякі відомості:

– T-Mobile мав дуже високий відсоток позитивних твітів;

– Verizon була єдиною компанією з більш негативними твітами, ніж позитивні;

– найпопулярніші ключові слова для позитивних твітів у Verizon включали типові терміни, такі як "новий телефон", "спасибі" і "якісне обслуговування клієнтів". Ключові пропозиції були типовими, формальними, трохи сухими взаємодіями між командою та послідовниками;

– найпопулярніші ключові слова для позитивних твітів у T-Mobile включали імена людей у своїй команді підтримки клієнтів, тому що їхня команда практикувала більш високу взаємодію, бесіди з клієнтами були більш персоніфікованими, тому користувачі отримували краще враження про взаємодію з цією компанією.

Підводячи підсумок, це може означати, що більш особистісне, залучення до соціальних медіа викликає більш позитивні відповіді та вищу задоволеність клієнтів. Тому для аналізу настроїв клієнтів доцільно використовувати інструменти аналізу тональностей тексту.

1.5.5 Аналіз настроїв у аналітиці робочої сили та голосу працівника

Так само, як ми вимірюємо VoC через опитування клієнтів, ми можемо вимагати і діяти на основі зворотного зв'язку від наших співробітників. Швидше за все, вони значно більше інвестують у надання дієвих ідей щодо вдосконалення робочого місця. І шанси, що ви, як роботодавець, дико більше зацікавлені в тому, щоб вони були залучені і наділені повноваженнями, щоб зробити все можливе.

Як можна використовувати аналіз настроїв:

– аналізуйте опитування співробітників, вибирайте ключові слова та переглядайте їх за сегментами;

– відстежувати зміни у настроях працівників у часі;

– вирішуйте поверхневі термінові проблеми негайно.

Аналіз настроїв корисний у аналітиці робочих місць і VoE, оскільки допомагає виконувати такі дії:

– відкрийте і вирішуйте проблеми співробітників, гарантуючи, що вони почують і цінують.

– розумійте VoE в реальному часі, а не щорічні огляди або огляди продуктивності.

Припустимо, що необхідно провести внутрішнє опитування, яке вимагає від співробітників оцінювати різні аспекти свого досвіду на робочому місці і пояснювати, чому вони так вважають. За шкалою від 1 до 10 найуспішніший працівник може сказати, що оцінює свою участь у роботі як 5 - не ідеально. Проте, якщо подивимося ближче, то побачимо, що вона додала: «Я люблю роботу, яку я роблю, і мої можливості для навчання були чудовими, але мій бос робить випадкові невідповідні зауваження до мене, які змушують мене відчувати себе незручно»

Відповідь, подібна до цього, повинна підняти червоні прапори потенційних сексуальних домагань і негайно привернути увагу персоналу, щоб вирішити ситуацію. Якщо просто скинути його разом з іншими сукупними оцінками і не читати їх ще два місяці, є ризик втратити цінного співробітника або підвищити вже напружену ситуацію.

1.5.6 Аналіз настроїв у аналітиці продукту

У нашому гнучкому світі ми дізналися, що продукти краще побудувати на ранніх етапах прототипування, часто вимагаючи зворотного зв'язку і продовжуючи повторювати і вдосконалювати. Але для багатьох виробничих команд, які вимагають частого зворотного зв'язку, може бути найбільш складною частиною. Як зменшити сегмент клієнта, який потрібно задати? Як ви сортуєте і

зважуєте всі їхні відгуки? Це саме те, де аналіз настроїв може змінити гру. Незалежно від того, аналізуючи опитування, взаємодії з клієнтами або соціальні медіа, машинне навчання дає змогу одразу оцінити величезну кількість відгуків про продукт.

Як можна використовувати аналіз настроїв:

– проаналізуйте велику кількість досліджень зі зворотним зв'язком з продуктом

– проаналізуйте всі соціальні медіа та онлайн-згадки про продукт

– фільтрувати коментарі по аспектах і по настроях, щоб побачити, що потрібно налаштувати і що потрібно зберегти.

– автоматично направляйте відповідні коментарі до груп продуктів.

Аналіз аналітичних даних корисний для аналітики продуктів, оскільки допомагає виконувати наступні дії:

– слід постійно переходити на вкладки того, що люблять люди та не люблять ваш продукт;

– встановіть ноль, у яких сегментах, які речі, і як звернутися до цих аудиторій;

– допоможіть команді розробників продуктів неймовірне розуміння особливостей продуктивності.

Приклад: MonkeyLearn

Команда проводить аналіз настроїв на взаємодію з клієнтською підтримкою та використовує ці знання для того, щоб розширити можливості кожного в нашій компанії, а не лише для наших агентів підтримки. Тому, коли клієнт згадує, що у них виникають труднощі з X або що вони хочуть бачити Y, ми надаємо цю інформацію безпосередньо людям, які створюють наші продукти та керують ними. Маємо реальний зворотний зв'язок з реальними клієнтами, безпосередньо досягаючи вух людей, до яких це мало найбільше. Як і будь-яка велика команда виробників, ми слухаємо клієнтів і задовольняємо їхні потреби. Занадто часто все, що потрібно, - це просто оснащення вашої команди правильним розумінням, що співпадає з клієнтом.

1.5.7 Аналіз настроїв у дослідженні та аналізі ринку

І в якості останнього випадку використання, аналіз настроїв надає переваги усім видам маркетингових досліджень і конкурентного аналізу. Незалежно від того, чи вивчає користувач новий ринок, передбачаєте майбутні тенденції або зберігаєте перевагу в конкуренції, аналіз настроїв може зробити всі зміни.

Як можна використовувати аналіз настроїв:

– проаналізуйте огляди продуктів вашого бренду та порівняйте їх з конкурентами;

– створюйте щотижневі, щомісячні або щоденні звіти - своєрідна система раннього попередження;

– порівняйте настрої на міжнародних ринках;

– проаналізуйте офіційні звіти ринку або бізнес-журнали для довгострокових, більш широких тенденцій;

– аналізуйте твіти та повідомлення соціальних медіа для подій у реальному часі;

– проаналізуйте відгуки про нефільтровані відгуки клієнтів;

– використовуйте аналіз на основі аспектів, що базується на аспектах, щоб отримати багате уявлення про деталі та причину непрозорих тенденцій на ринку.

Аналіз настроїв корисний для маркетингових досліджень та аналізу, оскільки допомагає:

– доторкніться до нових джерел інформації.

– кількісно визначити якісну інформацію.

– додайте цей якісний вимір до вже зібраних кількісних уявлень.

– надайте інформацію в реальному часі, а не в ретроспективі.

– автоматизація для регулярних (можливо, щотижневих) звітів.

– заповніть прогалини, де громадські дані є дефіцитними - наприклад, на ринках, що розвиваються.

Приклади: Відгуки про готелі на TripAdvisor.

Команда цікавилася тим, як люди відчувають себе в готелях у кількох великих містах світу, тому ми зібрали та проаналізували більше мільйона відгуків від TripAdvisor. Ми дивилися на готелі в Лондоні, Парижі, Нью-Йорку, Бангкоку, Мадриді, Пекіні та Ріо-де-Жанейро. В основному, відгуки були позитивними - у середньому 82% речей, які написали люди, позначені позитивним настроєм. Але готелі Лондона отримали найгірші відгуки. Лондон розглядався як більш брудний, ніж Нью-Йорк і з найгіршою їжею в цілому.

1.6. Постановка задачі розробки

На основі проведеного аналізу стану питання для визначення тональності текстів у магістерській кваліфікаційній роботі потрібно виконати такі задачі:

- розробити метод аналізу тональностей;
- розробити моделі систем автоматизованого визначення тональності тексту;
- розробити критерії автоматизованого визначення тональності тексту;
- провести навчання нейромережі;
- розробити програмне забезпечення системи визначення тональності текстів;
- провести тестування створеного програмного продукту.

1.7 Висновки

Здійснений у розділі аналіз можливостей використання технологій і методик аналізу тональності текстів переконливо доводить, що тематика нашого дослідження є актуальною в умовах сучасного розвитку суспільства і суспільних відносин. Така технологія дозволяє аналізувати настрої користувачів у соціальних мережах та медіаресурсах, швидкий аналіз і адекватна реакція на настрої користувачів дозволяє збільшити число відвідувачів, розширити коло тих, хто сприймає інформацію, а це означає, що адекватно реагуючи на настрої користувачів можна вести гнучку рекламну, політичну, соціальну, гендерну

політику й впливати при цьому на значну кількість населення. Аналіз настроїв у моніторингу брендів дозволяє швидко і адекватно реагувати на відношення користувачів до новинок на ринку і їх якості. Аналіз відгуків клієнтів дозволяє адміністрації закладів визначати відношення до працівників закладу, якості послуг, визначати потребу в їх розвитку тощо. Аналіз настроїв у службі підтримки та аналітиці робочої сили дозволяє оперативно реагувати на дії працівників офісу, розширювати коло потенційних клієнтів. Аналіз настроїв у аналітиці продукту та аналізі ринку дозволяє розширювати коло потенційних споживачів швидко і адекватно реагуючи на якість продукції.

2 РОЗРОБКА МЕТОДУ ТА МОДЕЛЕЙ СИСТЕМИ ВИЗНАЧЕННЯ ТОНАЛЬНОСТЕЙ ТЕКСТІВ

2.1 Аналіз технічного завдання

Автоматизована система визначення тональності тексту має бути орієнтована на роботу з веб-контентом та проводити ідентифікацію тональності текстових ресурсів з використанням методів машинного навчання.

Вхідні дані: веб-контент, відгуки користувачів.

Вихідні дані: встановлення тональності тексту.

Методи аналізу та ідентифікації тексту – для підвищення реалістичності результатів аналізу пропонується поєднати словникові, блочні методи ідентифікації текстів та машинне навчання, в основу якого покладено розроблені моделі автоматизованої системи аналізу тексту з використанням спеціалізованих нейромереж, призначених для веденя лінгвістично-емоційного аналізу текстових документів.

Враховуючи сучасні можливості нейронних мереж та машинного навчання, було прийнято рішення досліджувати методи аналізу тональностей тексту з використанням нейронних мереж. Отже, буде проведено дослідження можливостей використання різних типів архітектур та підходів у роботі з нейронними мережами для задачі аналізу тональностей текстів.

Великий об’єм інформації, з одного боку, дозволяє отримати більш точні розрахунки та здійснити точний аналіз, з другого – перетворює пошук рішень у складну задачу. Основна особливість Data Mining – це поєднання широкого математичного інструментарію та останніх досягнень у сфері інформаційних технологій, строго формалізованих методів та методів неформального аналізу, тобто кількісного та якісного аналізу даних. Класифікація методів інтелектуального аналізу даних представлена на рис. 2.1.

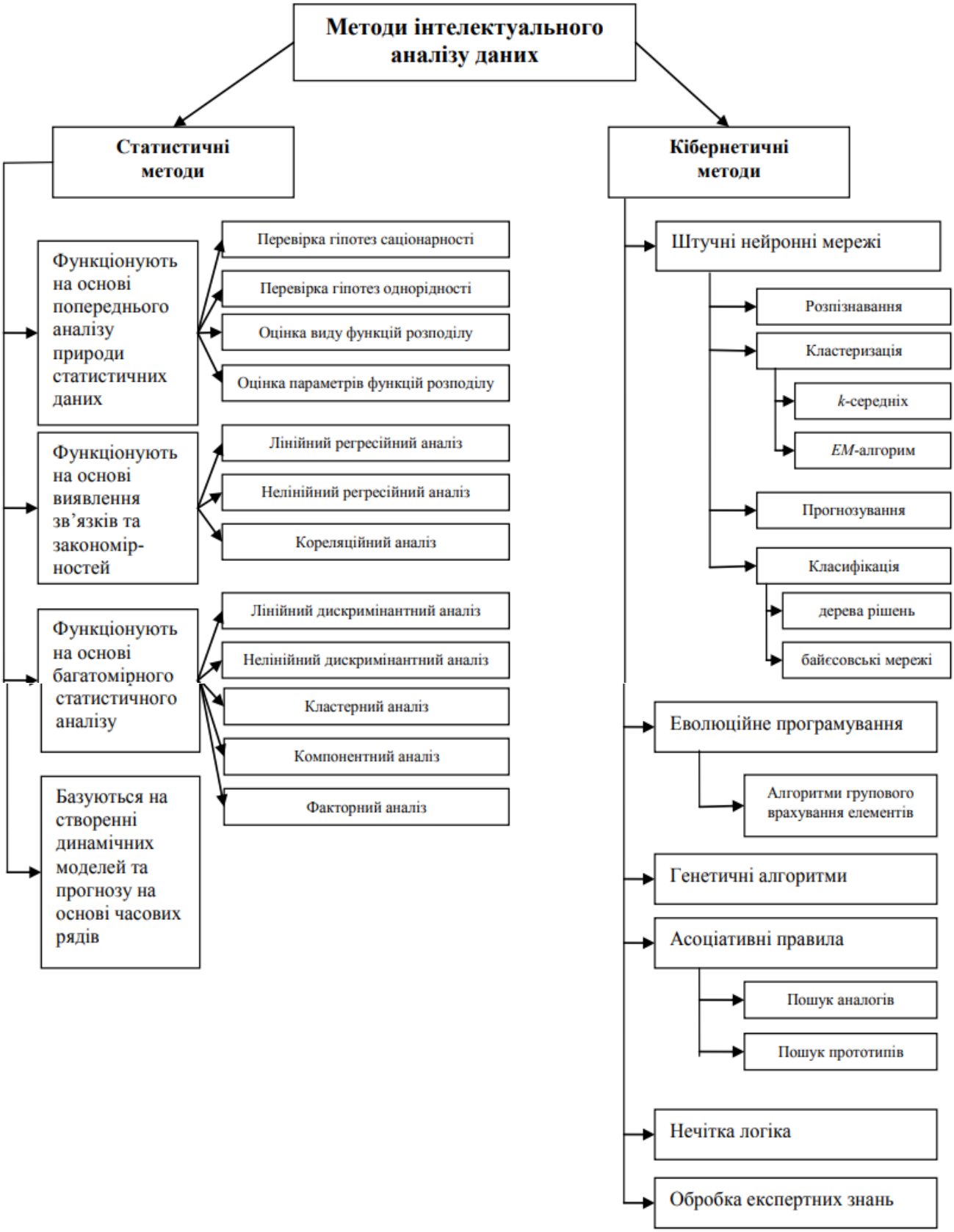


Рисунок 2.1 – Класифікація методів інтелектуального аналізу даних

2.2 Визначення критеріїв оцінки тональностей

2.2.1 Огляд датасетів

Ключовою частиною для освоєння аналізу настроїв є робота над різними наборами даних та різні експериментальні підходи. Для цього спочатку потрібно відібрати дані та отримати набір даних, над яким буде проводитися експеримент на основі визначеного домену та інтересів.

Розглянемо деякі з найбільш популярних наборів даних для аналізу експериментів з аналізом настроїв та підходу до машинного навчання (рис. 2.2). Вони відкриті та безкоштовні для завантаження [5].

Відгуки про продукт	набір даних, що складається з кількох мільйонів відгуків клієнтів Amazon із зірковими рейтингами, надзвичайно корисний для навчання моделі аналізу настроїв.
Відгуки ресторанів	набір даних складається з 5,2 мільйонів відгуків Yelp з рейтингом зірок.
Огляди фільмів	набір даних складається з 1000 позитивних і 1000 негативних оброблених відгуків, надає 5,331 позитивних і 5,331 негативних оброблених пропозицій / фрагментів.
Точні відгуки про продукти харчування	набір даних складається з ~ 500 тис. відгуків про продукти харчування від Amazon. Він включає в себе інформацію про продукт і користувача, рейтинги та звичайну текстову версію кожного огляду.
Відгуки авіакомпанії Twitter на Kaggle	набір даних складається з ~ 15000 позначених твітів (позитивний, нейтральний і негативний) про авіакомпанії.
Перший дебат у групі GOP Twitter Sentiment	набір даних складається з ~ 14 000 позначених твітів (позитивних, нейтральних та негативних) щодо першої дискусії GOP у 2016 році.

Рисунок 2.2 – найбільш популярні набори даних

Якщо користувачі зацікавлені в підході, що ґрунтується на правилах, нижче наведено різноманітний список лексиконів аналізу настроїв, які будуть корисними. Ці лексикони забезпечують набір словників слів з мітками, що визначають їхні настрої в різних галузях [4].

Розглянемо лексикони, що є дійсно корисними для визначення настрою текстів (рис. 2.3):

Рисунок 2.3 – лексикони для визначення настрою текстів

Лексикон AFINN є, мабуть, одним з найпростіших і найпопулярніших лексиконів, які можна широко використовувати для аналізу настроїв. Більш детальну і повну інформацію про цю лексику, розроблену та опрацьовану фіном Орупом Нільсеном, можна знайти у статті «Новий ANEW: оцінка списку слів для аналізу настроїв у мікроблогах». Цей лексикон у вільному доступі можна знайти

на офіційному сховищі GitHub автора разом з попередніми версіями його, включаючи AFINN-111.

2.2.2 Використання API для аналізу настроїв

API – це набір чітко визначених методів для взаємодії різних компонентів. API надає розробнику засоби для швидкої розробки програмного забезпечення.

Існує декілька варіантів систем аналізу настроїв, які можна використовувати через API. Загалом, їх можна розділити на дві різні категорії: бібліотеки з відкритим кодом та комерційні рішення. Відобразимо їх у моделі взаємозв'язку бібліотек ресурсів і комерційних рішень у процесі створення системи аналізу настрою (рис 2.4).

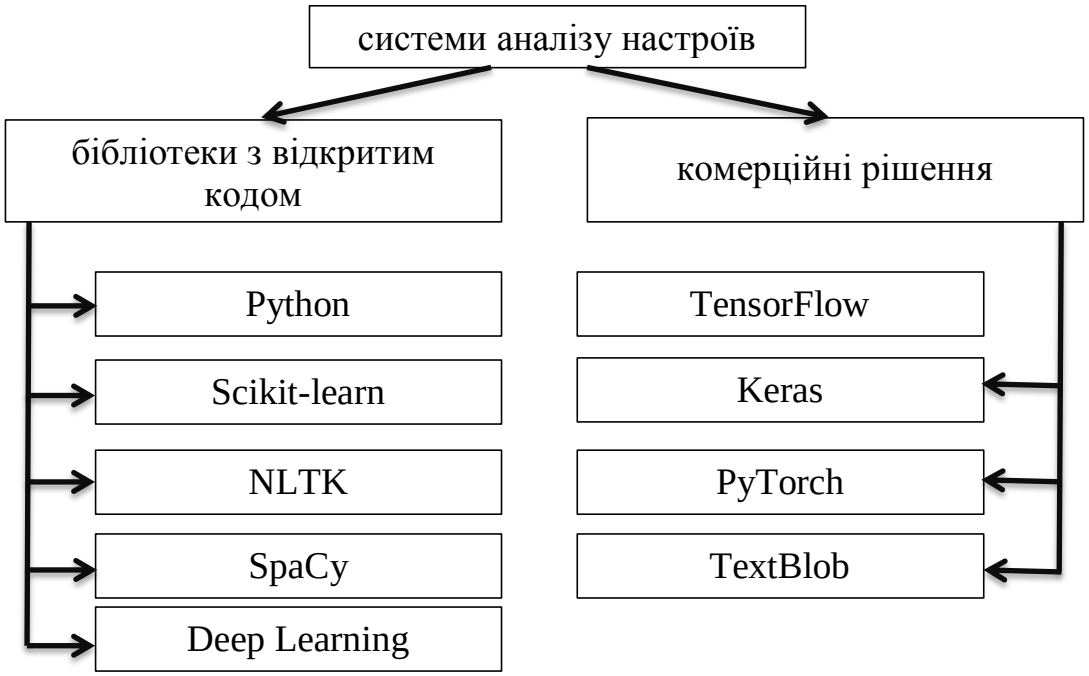


Рисунок 2.4 – моделі взаємозв'язку бібліотек ресурсів і комерційних рішень

Python є однією з провідних мов програмування для науки про дані, і має сильну спільноту і значний набір варіантів для реалізації моделей NLP.

Scikit-learn – це бібліотека для машинного навчання і корисні інструменти для векторизації тексту. Навчання класифікатору поверх векторизацій, таких як

частота або векторизатори тексту tf-idf, дуже проста. Scikit-learn має втілення для підтримки векторних машин, наївних байєсів і логістичної регресії.

NLTK – традиційна бібліотека для Python. Вона має активне співтовариство, і, крім того, що забезпечує низький рівень функцій для НЛП, вона також надає можливість тренувати класифікатори машинного навчання.

SpaCy є ще однією новою бібліотекою НЛП зі зростаючою спільнотою. Як і NLTK, він забезпечує сильний набір функцій низького рівня для НЛП і підтримку для підготовки класифікаторів тексту.

Завдяки тенденції Deep Learning в останні кілька років було розроблено новий набір бібліотек з інформатики, які підтримують застосування NLP.

TensorFlow – розроблений Google, він надає низькорівневий набір інструментів для створення та навчання нейронних мереж. Також існує підтримка векторизації тексту, як на традиційній частоті слова, так і на більш просунутих за допомогою вбудовування слів.

Keras надає корисні абстракції для роботи з декількома типами нейронних мереж, такими як рекурентні нейронні мережі (RNNs) і згорткові нейронні мережі (CNNs) і легко укладають шари нейронів [11]. Керас можна керувати на вершині Tensorflow або Theano. Він також надає корисні інструменти для класифікації тексту.

PyTorch – це новітня система глибокого навчання, яку підтримують деякі престижні організації, такі як Facebook, Twitter, Nvidia, Salesforce, Стенфордський університет, Оксфордський університет і Uber. Він швидко розвинув сильну громаду.

TextBlob – це ще одна чудова бібліотека з відкритим кодом для виконання завдань НЛП з легкістю, включаючи аналіз настроїв. Це також лексикон настрою (у вигляді XML-файлу), який він використовує, щоб дати оцінки полярності та суб'єктивності. Як правило, оцінки мають нормалізований масштаб порівняно з Afinn. Оцінка полярності являє собою поплавок у діапазоні [-1.0, 1.0]. Суб'єктивність є плаваючою в діапазоні [0.0, 1.0], де 0.0 дуже об'єктивно і 1.0 дуже суб'єктивно. Давайте скористаємося цим зараз, щоб отримати полярність і

мітки настроїв для кожної статті новин і об'єднати підсумкові статистичні дані для кожної категорії новин.

2.3 Розробка засобів визначення тональності тексту

2.3.1 Використання згорткових нейронних мереж

Згорткова нейронна мережа (convolutional neural network, CNN) – це клас нейронних мереж глибокого навчання, що зазвичай використовується для аналізу зображень, відео, а також обробки природних мов [12]. Згорткова нейронна мережа є регуляризованою версією багатошарового персептрона, який розроблено таким чином, щоб для роботи нейронної мережі потрібно було проводити мінімальну попередню обробку.

Робота згорткової нейронної мережі зазвичай є переходом від конкретних особливостей вхідних даних до більш абстрактних деталей, і далі до ще більш абстрактних деталей, доходячи до виділення понять високого рівня. Мережа є самоналаштованою і самостійно виробляє необхідну ієрархію абстрактних ознак чи послідовностей ознак, проводячи фільтрацію неважливих деталей і виділяючи важливі.

Ознаки, які виробляє нейронна мережа, зазвичай є доволі складними для розуміння, тому у випадку, якщо система ігнорує якісь істотні ознаки, замість зміни змісту ознак рекомендується удосконалити структуру та архітектуру мережі.

У звичайному перцептроні, який представляє собою повнозв'язну нейронну мережу, кожен нейрон пов'язаний з усіма нейронами попереднього шару, причому кожна зв'язок має свій персональний ваговий коефіцієнт [12].

У згортковій нейронній мережі в операції згортки використовується лише обмежена матриця ваг невеликого розміру, яку «рухаються» по всьому оброблюваного шару (на самому початку - безпосередньо по вхідним даним), де

формують після кожного зсуву сигнал активації для нейрона наступного шару з аналогічною позицією

Тобто для різних нейронів вихідного шару використовуються одна і та ж сама матриця ваг, яку також часто називають ядром згортки. Її інтерпретують як графічне кодування якої-небудь ознаки, наприклад, наявність похилої лінії під певним кутом чи наявність певних фігур, що наприклад повторюються на різних зображеннях (рис. 2.5). Тоді наступний шар, що вийшов в результаті операції згортки такою матрицею ваг, показує наявність даної ознаки в оброблюваному шарі і її координати, формуючи так звану карту ознак (англ. Feature map). При цьому такі ядра згортки не закладаються дослідником заздалегідь, а завжди формуються самостійно шляхом навчання мережі класичним методом зворотного поширення помилки.

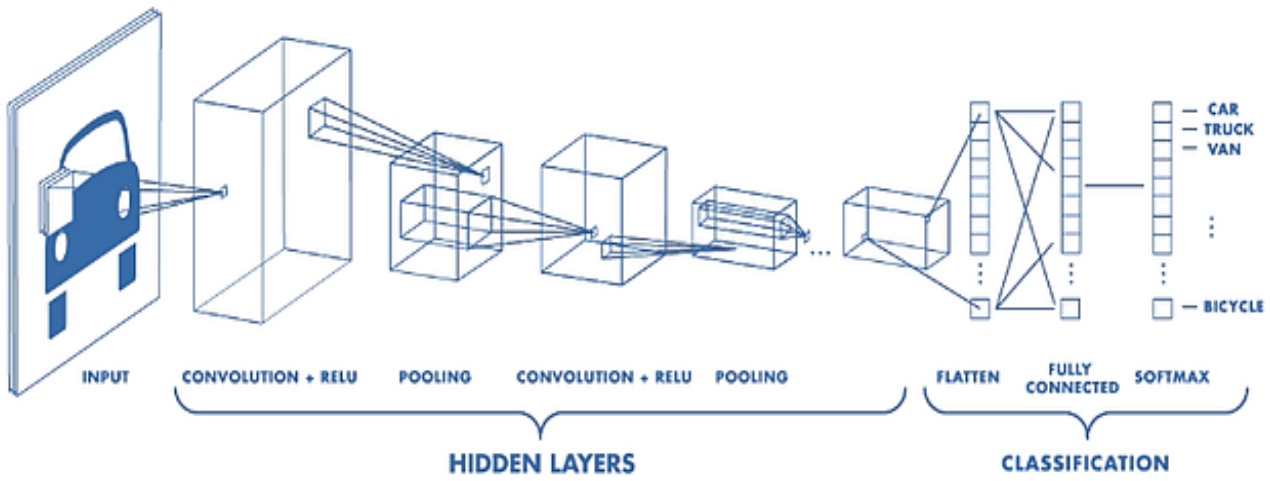


Рисунок 2.5 - Типова архітектура згорткової нейронної мережі

Природно, в згортковій нейронній мережі набір ваг не один, а ціла гама, що кодує елементи вхідних даних (наприклад лінії і дуги під різними кутами для зображення).

При цьому такі ядра згортки не закладаються дослідником заздалегідь, а формуються самостійно шляхом навчання мережі класичним методом зворотного поширення помилки.

Прохід кожним набором ваг формує свій власний примірник карти ознак, роблячи нейронну мережу багатоканальної (багато незалежних карт ознак на одному шарі). Також слід зазначити, що при переборі шару матрицею ваг її пересувають зазвичай не на повний крок (розмір цієї матриці), а на невелику відстань. Так, наприклад, при розмірності матриці ваг 5 × 5 її зрушують на один або два нейрона (пікселя) замість п'яти, щоб не «переступити» шукану ознаку.

Найбільш простим і популярним способом навчання є метод навчання з учителем (на маркованих даних) - метод зворотного поширення помилки і його модифікації.

Але існує також ряд технік навчання згорткової мережі без вчителя. Наприклад, фільтри операції згортки можна навчити окремо і автономно, подаючи на них вирізані випадковим чином шматочки вихідних зображень навчальної вибірки і застосовуючи для них будь-який відомий алгоритм навчання без вчителя (наприклад, автоассоціатор або навіть метод k-середніх) - така техніка відома під назвою patch-based training.

Відповідно, наступний шар згортки мережі буде навчатися на шматочках від уже навченого першого шару мережі. Також можна скомбінувати надточну нейромережу з іншими технологіями глибинного навчання. Наприклад, зробити згортковий авто-асоціатор, згорткову версію каскадних обмежених машин Больцмана, що навчаються за рахунок імовірнісного математичного апарату, згорткову версію розрідженого кодування (англ. Sparse coding), названу deconvolutional networks («розгорткові» мережі) [13].

Для проведення дослідження роботи згорткової нейронної мережі буде використовуватися Keras. Це відкрита бібліотека для роботи з нейронними мережами, яка містить необхідні методи для обробки вхідних даних у необхідний для роботи мережі вигляд.

2.3.2 Застосування рекурентної нейронної мережі

Рекурентна нейронна мережа (Recurrent neural network; RNN) – клас нейронних мереж, в якому міжвузлові з’єднання утворюють направлену у часі

послідовність (орієнтований граф) [14]. Завдяки цьому рекурентні нейронні мережі надають можливість опрацьовувати послідовні просторові ланцюги чи серії подій у часі.

Рекурентні нейронні мережі (англ. Recurrent Neural Networks, RNN) – це мережі, що містять зворотні зв’язки і дозволяють зберігати інформацію (рис. 2.6).

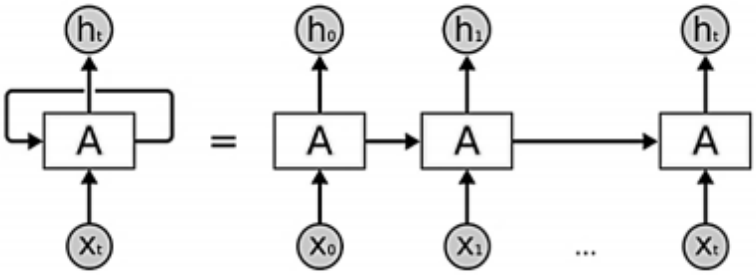


Рисунок 2.6 - Рекурентна мережа у розгортці

На схемі вище фрагмент нейронної мережі A приймає вхідний значення x_t і повертає значення h_t. Наявність зворотного зв’язку дозволяє передавати інформацію від одного кроку навчання мережі до іншого.

Рекурентні мережі, на відміну від багатошарових перцептронів, можуть використовувати свою внутрішню пам’ять для обробки послідовностей довільної довжини. Через це подібні нейронні мережі є широко застосовними в задачах розпізнавання тексту чи мови, а також для задач обробки природних мов [15], де вони дозволяють широко використовувати попередній досвід.

Найбільш розповсюдженими є варіанти архітектури рекурентної нейронної мережі з довгою короткочасною пам’яттю (LSTM) та керованим рекурентним блоком (GRU).

У дослідженні буде використано варіант архітектури з довгою короткочасною пам’яттю.

Одним з різновидів RNN є LSTM-мережі. LSTM (англ. Long Short Term Memory) – це RNN здатні до навчання довготривалими залежностями. LSTM-мережі складаються з повторюваних елементів. Кожен такий елемент містить чотири шари і відрізняється тим, що має комірку довгої короткочасної пам’яті (рис. 2.7).

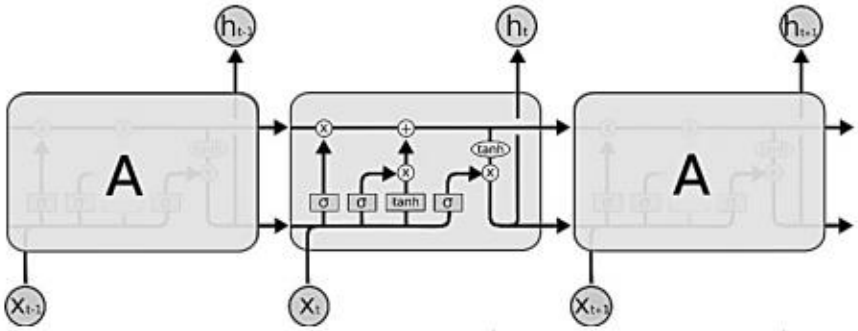


Рисунок. 2.7 – Повторюваний модуль в LSTM, що містить чотири взаємодіючих шари

LSTM-мережа добре пристосована для задач з класифікації [16]. Ця штучна нейронна мережа містить LSTM-модулі замість або в якості доповнення до інших мережевих модулів.

LSTM-модуль це рекурентний модуль мережі, який здатен запам'ятовувати значення як на короткі, так і на довгі проміжки часу. LSTM-модуль не використовує функцію активації в середині своїх рекурентних модулів, і значення, що зберігається, не розмивається у часі, тому градієнт не зникає, коли використовується метод зворотного розпізнавання розповсюдження помилки у часі під час тренування мережі.

LSTM-модулі часто складаються у блоки. Таке будування характерно для глубоких багатошарових нейронних мереж і дозволяє використовувати паралельні обчислення та спеціальне обладнання.

LSTM-блоки містять три або чотири "вентилі" (gates), що використовуються для контролю інформації на входах та виходах пам'яті цих блоків (рис. 2.8).

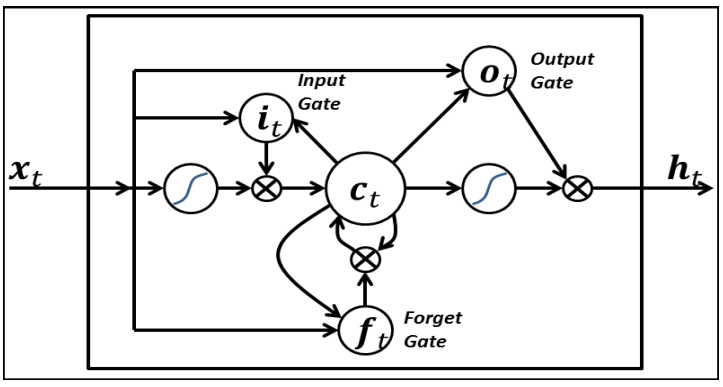


Рисунок 2.8 - LSTM-блок з трьома вентилями

Ці вентилі реалізовані у вигляді логістичної функції для обчислення значень в діапазоні [0; 1]. Множення на це значення використовується для часткового допуску чи заборони потока інформації в середину або назовні пам’яті[17]. Наприклад, «вхідний вентиль» (input gate) керує мірою входження нового значення у пам’ять, «вентиль забуття» (forget gate) керує тим, до якої міри значення залишається в пам'яті. А «вихідний вентиль» (англ. output gate) керує тим, до якої міри значення в пам'яті використовується для обчислення активації виходу блоку.

Ваги в LSTM-блоці (W і U) використовуються для завдання напрямку оперування вентилів. Ці ваги визначені для значень, які подаються в блок (x_t і вихід з попереднього тимчасового кроку h_{t-1}) для кожного з вентилів. Таким чином, LSTM-блок визначає, як розпоряджатися своєю пам'яттю як функцією цих значень, і тренування ваг дозволяє LSTM-блоку вивчити функцію, що мінімізувала втрати. LSTM-блоки зазвичай тренують за допомогою методу зворотного поширення помилки в часі.

LSTM-мережі застосовують для керування роботами, розпізнавання мовлення, навчання граматики, розпізнавання дій людей, прогнозування часових рядів, навчання ритму, виявлення гомології білків, навчання ритму[18].

2.3.3 Використання попередньо навчених моделей нейронних мереж

Попередньо підготовлена модель – це збережена мережа, яка раніше навчалася на великому наборі даних, як правило, на масштабному завданні класифікації зображень. У цьому підході або проходить використання моделі, яка вже була оброблена, або використовується трансферне навчання, щоб налаштувати цю модель для даного завдання.

Трансферне навчання - це методика машинного навчання, де модель, що навчається на одному завданні, переорієнтована на друге відповідне завдання. Згідно з означенням з книги Deep Learning[19], трансферне навчання та адаптація домену відносяться до ситуації, коли те, що було вивчено в одній установці

використовується для поліпшення узагальнення в іншому середовищі. Це вдосконалення навчання в новому завданні через передачу знань з пов'язаного завдання, яке вже було вивчено.

Ідея трансферного навчання полягає в тому, що якщо модель навчена на великому і загальному наборі даних, то ця модель буде ефективно служити загальною моделлю візуального світу. Потім є можливість скористатися цими вивченими картами властивостей без необхідності починати з нуля навчання великої моделі на великому наборі даних.

В останні роки глибоке навчання досягло значного прогресу. Це дозволило нам вирішувати складні проблеми і давати дивовижні результати. Однак час навчання та обсяг даних, необхідних для таких систем глибокого навчання, набагато більше, ніж у традиційних систем ML. Існують різні глибокі мережі навчання з найсучаснішими діями (іноді так добре або навіть краще, ніж продуктивність людини), які були розроблені та випробувані в різних областях, таких як комп'ютерне бачення та обробка природної мови [20]. У більшості випадків команди/користувачі діляться деталями цих мереж, щоб інші могли їх використовувати. Ці попередньо навчені мережі/моделі формують основу трансферного навчання в контексті глибокого навчання, або те, що можна називати "глибоким трансферним навчанням".

Для проведення аналізу тональностей тексту можуть використовуватися багатоцільові моделі для обробки природної мови. Ці моделі підсилюють застосування для обробки природних мов - машинний переклад, системи відповіді на запитання, чат-боти, аналіз настроїв тощо. Основний компонент цих багатоцільових моделей обробки природних мов – концепція мовного моделювання.

Найвідомішими є наступні попередньо треновані моделі:

– ULMFiT – модель, натренована на наборі даних з сайту Wikitext, завдяки чому модель може використовуватися для широкого спектру задач обробки природної мови;

– Transformer – модель, представлена компанією Google, для тренування якої були використані згорткові та рекурентні нейронні мережі;

– Google’s BERT – інша попередньо натренована модель від Google, яка містить результати виконання 11 задач обробки природної мови, в тому числі аналіз тональностей тексту[21];

– ELMo – модель, що використовує вбудовування слів (англ. Word Embeddings) – перетворення текстових даних у числові.

Для проведення дослідження буде використовуватися модель Google’s BERT.

BERT – це метод попередньої підготовки мови, що означає, що тренується загальна модель "розуміння мови" на великому текстовому корпусі (наприклад, з Вікіпедії), а потім ця модель використовується для задач обробки природної мови,. BERT перевершує попередні методи, оскільки це перша без нагляду, глибоко двонаправлена система для попереднього навчання обробки природної мови [21].

У даному контексті характеристика «Без нагляду означає», що BERT навчався, використовуючи лише звичайний текст, що є важливим, оскільки величезна кількість звичайних текстових даних є загальнодоступною в Інтернеті багатьма мовами.

Попередньо навчені представлення також можуть бути контекстно-вільними або контекстуальними, а контекстні подання можуть бути однонаправленими або двонаправленими. Контекстні моделі, такі як word2vec або GloVe, створюють єдине "вбудовування слова" для кожного слова у словнику, тому банк матиме таке ж уявлення в банківському депозиті та річковому березі. Контекстні моделі замість цього створюють представлення кожного слова, яке базується на інших словах у реченні.

2.4 Розробка методу визначення тональності текстів

2.4.1 Розробка методу та моделі системи визначення тональності тексту

Для аналізу тексту, що є однією з основних задач обробки природної мови (рис. 2.9), сьогодні використовуються точні методи, які спрямовані на розуміння текстів та екстракцію знань, що передбачає розбиття текстів на речення, вилучення об’єктів, синтаксичний аналіз та ін.

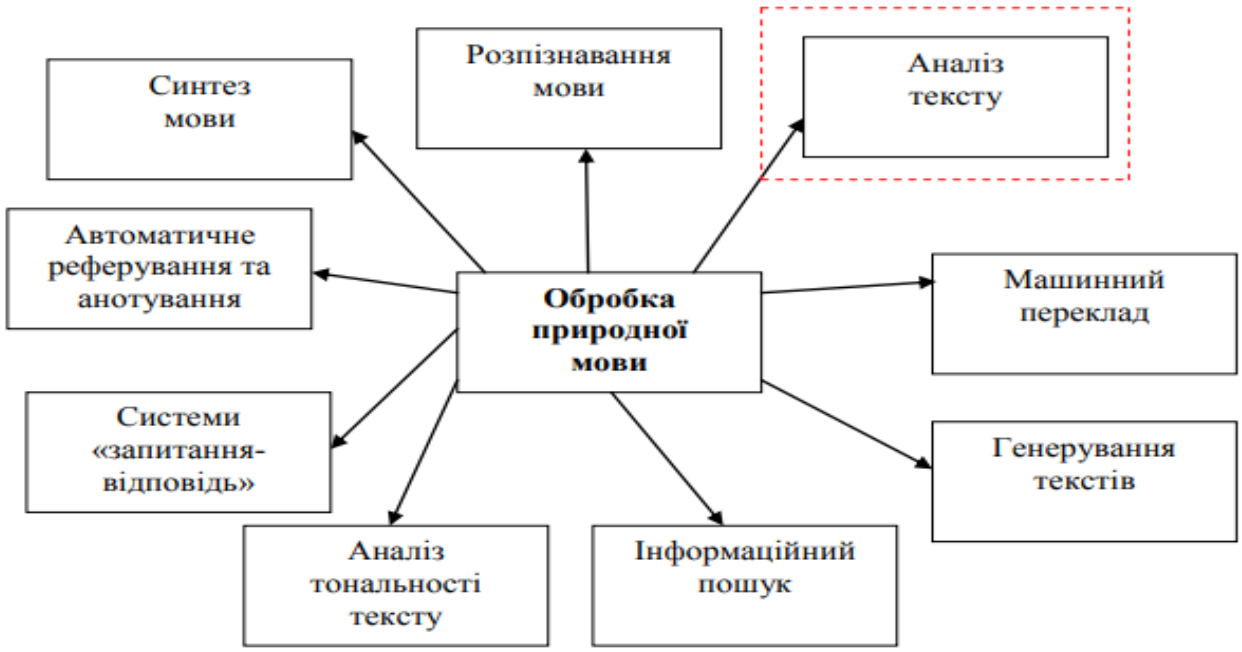


Рисунок 2.9 – Основні задачі NLP

В інтелектуальному аналізі даних для виявлення закономірностей та тенденцій, які існують між даними та які неможливо виявити при традиційному перегляді через їх складність, застосовується математичний аналіз. Загалом, закономірності та тренди визначають модель інтелектуального аналізу даних, яку можна застосовувати для прогнозування, розрахунку ризиків та ймовірностей, висунення рекомендацій, пошуку послідовностей, групування та ін. При цьому побудова моделі є частиною масштабного процесу (рис. 2.10).

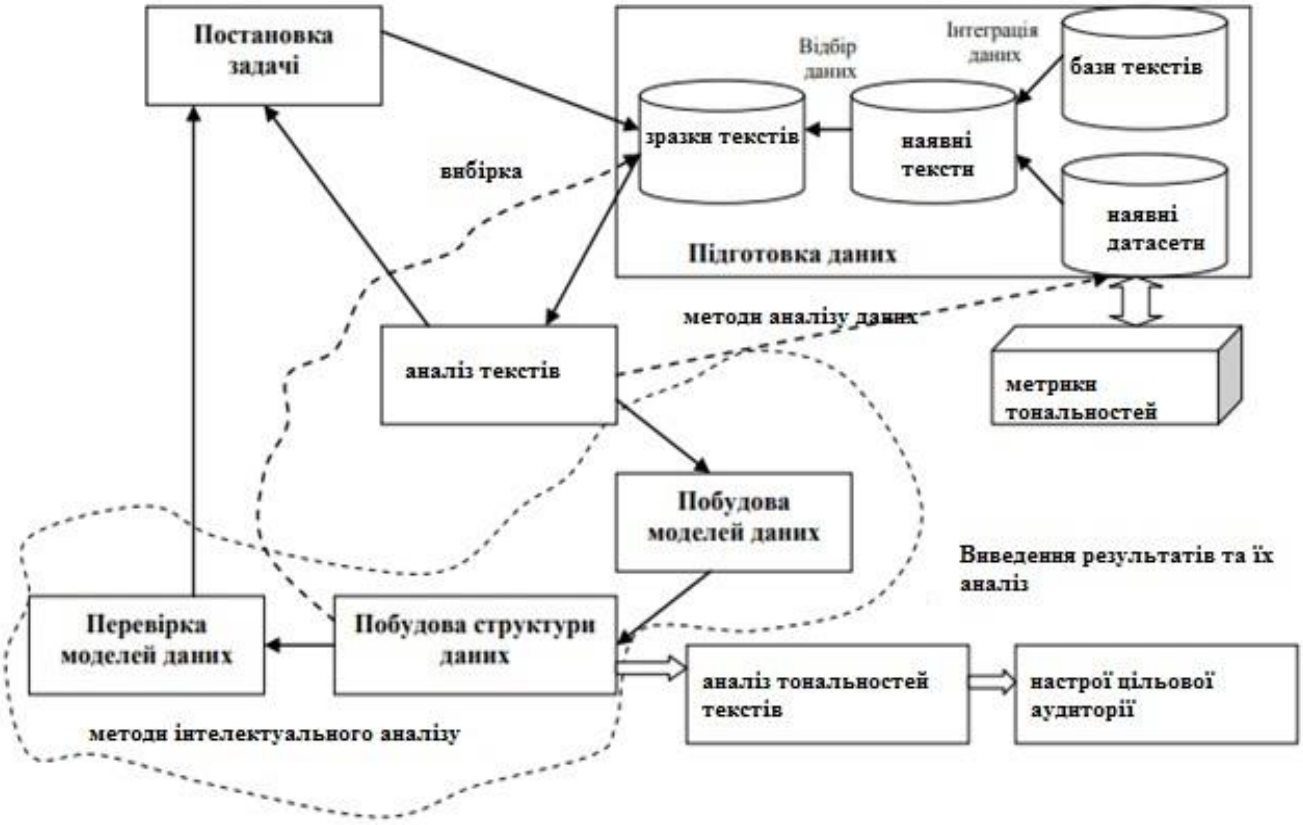


Рисунок 2.10 – Загальна модель процесу інтелектуального аналізу даних

Метод визначення тональностей складається з таких етапів:

- отримання доступу до текстових даних, які будуть подані як вхідні дані для аналізу та ідентифікації (контент сайту та відгуки користувачів);
- оцінка тексту за критеріями відбору тональностей;
- визначення ключових слів та інформаційних блоків як вхідних даних текстового класифікатора;
- ідентифікація типів тональностей з використанням методів машинного навчання та нейромереж;
- візуалізація результатів ідентифікації тональності тексту.

Дослідження буде проведено на наборі даних IMDB Movie Reviews Dataset [10]. Цей набір даних містить 50000 відгуків на фільми з сайту Internet Movie Database – це найбільша в світі база даних та веб-сайт про кінематограф, який містить не лише вичерпну інформацію про фільми, серіали та ін., а також дає можливість користувачам ставити оцінки фільмам та писати до них відгуки. Саме

ці відгуки буде використано для проведення дослідження. Набір даних розділено на два набори: 25000 для тренувального набору та 25000 для тестувального набору. Відгуки у даному наборі розділені лише на два типи: негативні та позитивні (набір містить рівну кількість позитивних та негативних відгуків), тобто буде проводитися бінарна класифікація.

2.4.2 Проведення оцінки тексту за критеріями відбору тональностей з допомогою нейронних мереж

Для проведення дослідження були обрані такі класи нейронних мереж та підходи:

– згорткова нейронна мережа (convolutional neural network, CNN);

– довга короткочасна пам’ять (long short-term memory, LSTM), варіант архітектури рекурентної нейронної мережі;

– використання попередньо навченої моделі;

Результатом проведеного дослідження буде порівняння точності класифікації відгуків як позитивних чи негативних для трьох зазначених вище підходів. Мірою точності буде виступати відсоток правильних передбачень з 25 тисяч відгуків.

Також буде розроблено веб-додаток, у якому користувач матиме можливість визначити емоціональне забарвлення довільного тексту, використовуючи три варіанти попередньо тренованих нейронних мереж: згорткова нейронна мережа, рекурентна нейронна мережа (архітектура коротка довгочасна пам’ять), та модель Google’s BERT (опис кожного підходу буде наведено нижче.

2.4.3 Визначення ключових слів, словосполучень, фраз

Створення програмного продукту для аналізу тональності тексту розбивається на кілька етапів:

1) Імпорт документа в форматі .doc, .docx, .pdf, .odt в текстовий формат з видаленням непотрібної інформації (зображень, розмітки, гіперпосилань і т. д.)

2) Створення текстового класифікатора.

I) Виділення бази однокореневих слів для приведення їх до однаковому виду ("стемінг")

II) Створення матриці документів. У цій матриці кожен з відгуків - рядок, а стовпець - з усього того проаналізованих відгуків. У таблиці ставиться 0, якщо це слово не зустрічається в відкликання і 1, якщо зустрічається.

III Виділення найбільш значущих слів, які найсильніше впливають на рейтинг відкликання (цей крок необхідний, тому що таблиця виходить дуже великий)

IV) Аналіз тексту на основі алгоритму логит-регресії

Для вилучення тексту з файлів використовуються спеціальні

бібліотеки:

Формат	Библиотека	Лицензия
.doc	NPOI	Apache
.docx	SDK from Microsoft	Proprietary
.pdf	PDFBox	BSD
.odt	AODL	Apache

2.4.4 Ідентифікація типів тональностей тексту

Всі ліцензії, під якими поширюються дані бібліотеки, дозволяють використовувати бібліотеки безкоштовно, в тому числі і для комерційного використання.

Перед будь-обробкою даних потрібно попередня обробка. У цій стадій видаляються всі html теги, пунктуаций, символи. Дана операція здійснюється за допомогою бібліотеки python - «Beautiful Soup». Також всі числа і посилання в тексті замінюються на теги,. Далі в тексті присутні так звані «стоп слова» - це часті слова в мові, які в основному не несуть ніякої смислове навантаження (наприклад, в англійській мові це такі слова як «the, at, about ...»). Стоп слова видаляються за допомогою пакета Python Natural Language Toolkit (NLTK). Після

попередньої обробки вихідного тексту отримуємо наступне: [Biography, part, feature, film, remember, going, see, cinema, originally] - Тобто набір слів.

2.5 Висновки

Проведений аналіз методів дослідження дозволив визначити необхідні для розробки програми методи аналізу інтонацій тексту, визначити їх переваги і недоліки з метою використання визначених переваг у розробленій системі ідентифікації тональностей тексту.

Розроблено засоби визначення тональностей з використанням згорткових нейронних мереж та попередньо навчених моделей.

Розроблено метод та модель системи ідентифікації тональностей тексту.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ СИТСЕМИ ВИЗНАЧЕННЯ ТОНАЛЬНОСТЕЙ ТЕКСТІВ

3.1 Обґрунтування вибору засобів реалізації програми

У якості основної мови розробки для проведення дослідження було обрано мову програмування Python.

Python є інтерпретованою мовою програмування високого рівня загального призначення. Створений Гвідо ван Россумом і вперше випущений в 1991 році, Його мовні конструкції та об'єктно-орієнтований підхід спрямовані на те, щоб допомогти програмістам написати чіткий, логічний код для малих і великих проектів.

Наразі Python широко використовується для виконання задач машинного навчання та наукових розрахунків. Простота синтаксису та відносно невисокий поріг входу роблять його дуже привабливим для проведення досліджень як без, так і з використанням машинного навчання та нейронних мереж. Пайтон є кросс-платформовим, тому може використовуватися під різними операційними системами, під нього створені дуже зручні середовища розробки, як безкоштовні, так і комерційні. Окрім цього, для мінімального використання Пайтону не потрібні навіть середовища розробки.

Але найбільшою перевагою Python є велика кількість бібліотек, що можуть бути використані для наукових досліджень з використанням машинного навчання та нейронних мереж. Використані в дослідженні бібліотеки та платформи буде описано нижче.

Для роботи з згортковими нейронними мережами буде використовуватися TensorFlow. TensorFlow - це відкрита платформа з відкритим вихідним кодом для машинного навчання. Вона має всеосяжну, гнучку екосистему інструментів, бібліотек та ресурсів спільноти, що дозволяє дослідникам підштовхувати сучасні технології розробки ML, а розробники легко створюють та розгортають додатки, що працюють на основі ML.

Для роботи з TensorFlow буде використовуватися Keras.

Keras - це високоякісний API для створення моделей глибокого навчання[15]. Він використовується для швидкого створення прототипів, складних досліджень, а також для створення додатків. Три ключових переваги Keras API:

– простота в використанні: Keras має простий інтерфейс, оптимізований для більшості розповсюджених задач глибокого навчання. Також він дає конкретні підказки як швидко виправити можливі помилки;

– модульність: моделі Keras будуються за допомогою об'єднання декількох простих модулів, кожен з яких може бути налаштований незалежним чином;

– легко розширити модель: користувач може створювати власні модулі, необхідні для конкретного дослідження.

Для роботи з датасетом буде використовуватися бібліотека NumPy.

NumPy є основним пакетом для наукових обчислень з Python. Він містить, серед іншого:

– потужний об'єкт N-розмірного масиву;

– складні широкоформатні функції;

– інструменти для інтеграції C / C ++ і Fortran;

– корисна лінійна алгебра, перетворення Фур'є і можливості випадкових чисел.

Крім очевидного наукового використання, NumPy також може використовуватися як ефективний багатовимірний контейнер загальних даних. Можуть бути визначені довільні типи даних. Це дозволяє NumPy легко і швидко інтегруватися з широким спектром баз даних.

Для роботи з числовими таблицями буде використовуватися бібліотека Pandas. Це програмна бібліотека мови Python для обробки та аналізу даних. Работа панд з даними будується поверх бібліотеки NumPy, що є інструментом більш низького рівня. При наявності пакету matplotlib ця бібліотека дає можливість малювати графіки на отриманих наборах даних.

Для роботи з LTSM та попередньо тренованою моделлю буде використовуватися PyTorch. Це бібліотека машинного навчання для мови Python з

відкритим вихідним кодом, створена на базі Torch [18]. Пакет torch реалізує основну структуру даних бібліотеки – n-мірний тензор, а також базові методи для роботи з ним – стандартні математичні та статистичні операції, базові підпрограми лінійної алгебри. Використовується для обробки природної мови.

Для реалізації користувацького інтерфейсу використовується бібліотека React, одна з сучасних JavaScript-бібліотек з відкритим вихідним кодом для розробки користувацьких інтерфейсів.

Основним середовищем розробки обрано PyCharm від JetBrains. Це інтегроване середовище розробки для мови програмування Python. Надає засоби для аналізу коду, графічний відладчик, інструмент для запуску юніт-тестів і підтримує веб-розробку на Django.

У якості платформи використовується Windows.

3.2 Розробка модуля машинного навчання із застосуванням згорткових нейронних мереж

Розглянемо детальніше етапи, починаючи з приведення до довжини.

Конвертація відгуків в тензори може бути зроблена декількома способами.

One-hot encoding конвертує масиви в вектори 0 і 1. Наприклад, послідовність [3, 5] стане 10,000-мірним вектором, повністю складається з нулів крім показників 3 і 5, які будуть представлені одиницями. Потім, нам потрібно буде створити перший Dense шар в нашій мережі, який зможе приймати вектор дані з плаваючою комою. Такий підхід дуже вимогливий до обсягу пам'яті, незважаючи на те, що вимагає вказати розміри матриці num_words * num_reviews.

Інший спосіб - зробити все масиви однаковими по довжині, а потім створити тензор цілих чисел із зазначенням max_length * num_reviews. Ми можемо використовувати Embedding (пер. "Вбудований") шар, який може використовувати ці параметри в якості першого шару нашої мережі.

Обрано було саме другий варіант, бо він менш вибагливий до обсягу пам’яті. Тому першим етапом проведено приведення відгуків до загальної довжини в 256 слів. Це можна зробити за допомогою вбудованої функції pad_sequences. Одразу обрізаємо і тренувальні, і тестові відгуки. Зразок програмного коду для здійснення цієї операції надано нижче.

```
    train_data =
keras.preprocessing.sequence.pad_sequences(train_data,
value=word_index["<PAD>"],
    padding='post',
    maxlen=256)
    test_data = keras.preprocessing.sequence.
pad_sequences(test_data,
    value=word_index["<PAD>"],
    padding='post',
    maxlen=256)
```

Наступний етап – власне побудування моделі.

Для створення класифікатора всі верстви проходять процес стека, або накладення:

– перший Embedding шар приймає перекладені в цілі числа слова і шукає відповідний вектор для кожної пари слово/число. Модель навчається на цих векторах. Вектори збільшують розмір одержуваного масиву на 1, в результаті чого ми отримуємо вимірювання: (batch, sequence, embedding);

– наступний шар GlobalAveragePooling1D повертає отриманий вектор заданої довжини для кожного прикладу, усереднюючи розмір ряду. Це дозволить моделі легко приймати дані різної довжини;

– цей вектор пропускається через повнозв'язний Dense шар з 16 прихованими блоками;

– останній шар також є повнозв'язним, але з усього одним вихідним вузлом. За допомогою функції активації sigmoid (сигмоид) отримується число з плаваючою комою між 0 і 1, яке буде показувати ймовірність або впевненість моделі.

Нижче наведено приклад програмного коду для побудови моделі нейронної мережі:

```
    vocab_size = 25000
    model = keras.Sequential()
    model.add(keras.layers.Embedding(vocab_size, 16,
input_shape=(None,)))
    model.add(
    keras.layers.GlobalAveragePooling1D())
    model.add(keras.layers.Dense(16,
activation=tf.nn.relu))
    model.add(keras.layers.Dense(1,
activation=tf.nn.sigmoid))
    model.summary()
```

Вищеописана модель має 2 проміжних або приховані прошарки, між входом і виходом даних. Кількість виходів (блоків, нодів або нейронів) є розміром репрезентативного простору шару. Іншими словами, кількість свободи, яка дозволена мережі під час навчання.

Якщо модель має більше прихованих блоків, і/або більше шарів, то тоді нейромережа може навчитися більш складним уявленням. Однак в цьому випадку це буде дорожче з точки зору обчислювальних ресурсів і може призвести до навчання небажаних патернів – патернів, які покращують показники на тренувальних даних, але не на перевірочних. Це називається перенавчанням.

Для моделі необхідно вказати функцію втрат і оптимізатор для навчання. Оскільки розв’язувана задача є прикладом бінарної класифікації та модель буде показувати ймовірність (шар з єдиного блоку з сигмоид як функції активації), то буде використовуватися функція втрат binary_crossentropy (пер. "Перехресна ентропія").

Це не єдиний вибір для функції втрат: можна, наприклад, вибрати mean_squared_error. Але зазвичай binary_crossentropy краще справляється з вірогідністю – вона вимірює "дистанцію" між розподілами ймовірностей, або, як у нашому випадку, між еталоном і прогнозами.

При подальшому налаштуванні моделі використовується оптимізатор Адама і перехресна ентропія для втрат:

```
model.compile(
              optimizer=tf.train.AdamOptimizer(),
              loss='binary_crossentropy',
              metrics=['accuracy'])
```

Модель налаштована, і далі проводиться тренування моделі на тренувальній частині датасету, яка містить двадцять п’ять тисяч відгуків, порівну негативних та позитивних.

3.3 Розробка модуля ідентифікації тональності даних із застосуванням LTSM нейронної мережі

Для роботи з LTSM нейронною мережею використовується не готовий набір даних, підготовлений для тренування та тестування нейронної мережі, тому до етапів дослідження додаються початкові етапи підготовки даних.

Загалом дослідження складається з наступних етапів:

– обробка даних – перетворення в нижній регістр, видалення розділових знаків;

– токенізація – створюється список слів за частотою використовування, кожне слово у відгуку замінюється на відповідне число (те є саме, що робилося для попереднього етапу дослідження);

– аналіз довжини відгуків, приведення усіх відгуків до спільної середньої довжини;

– визначення мережевої архітектури LSTM;

– побудування класу моделі;

– навчання мережі;

– тестування мережі.

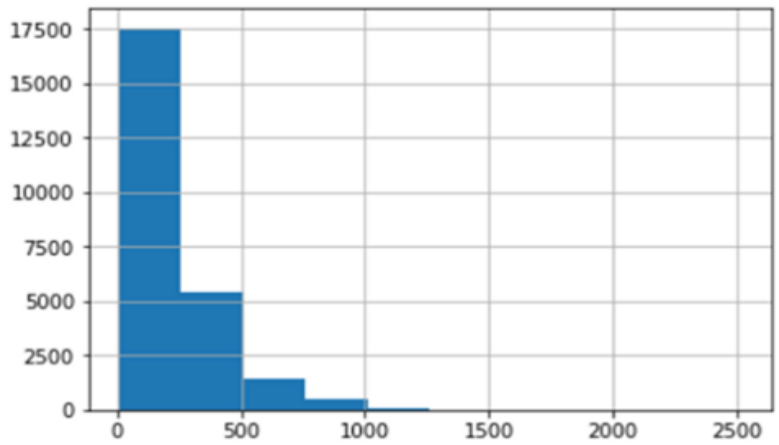
Розглянемо детальніше етапи в-е, оскільки перші два етапи є такими ж самими, як для попереднього дослідження.

Для того, щоб визначити середню довжину відгуків, використовується бібліотека pandas. Нижче зображено приклад програмого коду для виявлення

середньої довжини відгуків та наочного відображення отриманих даних у вигляді гістограми.

```
import pandas as pd
reviews_len =
    [len(x) for x in reviews_int]
pd.Series(reviews_len)
    .hist()
plt.show()
pd.Series(reviews_len)
    .describe()
```

Для більшої наочності результатів побудована гістограма (див. рис. 3.6). На осі абсцисс зображено кількість слів, а а осі ординат - кількість відгуків, що мають довжину у заданому проміжку з кроком в 250 слів.




```
count    25001.000000
mean       240.798208
std        179.020628
min          0.000000
25%        130.000000
50%        179.000000
75%        293.000000
max       2514.000000
dtype: float64
```

Рисунок 3.6 – Гістограмма середньої довжини відгуків

Отже, як можна побачити з результатів, середня довжина відгуків – 241 символ. Як і у минулому дослідженні, відгуки приводяться до загальної довжини. Ці перетворення проводяться на обох наборах даних: на тренувальному та на тестовому. В результаті ми маємо два набори відгуків однакової довжини, в яких слова закодовані цифрами. Таким чином, дані підготовлені для подальшого використання нейронною мережею.

Далі будується власне нейронна мережа та клас моделі. Спочатку, визначаються наступні гіперпараметри:

– lstm_size: Кількість одиниць у прихованих шарах у клітинах LSTM. Зазвичай більше значення краще с точки зору ефективності. Загальні значення 128, 256, 512 і т.д., але використовується середня довжина відгуку;

– lstm_layers: кількість шарів LSTM в мережі. Я б почав з 1, а потім додавав більше, якщо я недостатньо підібраний;

– batch_size: кількість відгуків для передачі мережі в одному навчальному пропуску. Як правило, це має бути встановлено настільки високим, як можна перейти без пам'яті;

– learning_rate: швидкість навчання.

Обрано наступні параметри:

```
lstm_size = 256
lstm_layers = 2
batch_size = 1000
learning_rate = 0.01
```

В загальному вигляді, архітектура нейронної мережі складається з наступних шарів:

– embedding шар, що перетворює слова (цілі числа) у вектори певного розміру;

– шар LSTM: визначається вимірами схованого стану і кількістю шарів;

– повністю підключений шар: що відображає вихід LSTM-шару на бажаний розмір виводу;

– рівень активації сигмоїдів: перетворює всі вихідні значення у значення від 0 до 1

– вихідні дані: Сигмоподібний вихід з останнього тимчасового кроку розглядається як кінцевий вихід цієї мережі.

Далі визначається клас моделі (повний код класу моделі розміщено у додатках), після чого проходить створення інстанції нейронної мережі і проходить тренування на тренувальних даних. Після закінчення тренування проводиться перевірка на тестових даних.

```
test_acc = []
with tf.Session() as sess:
```

```
        saver.restore(sess,
"checkpoints/sentiment_manish.ckpt")
        test_state = sess.run(cell.zero_state(batch_size,
tf.float32))
        for ii, (x, y) in enumerate(get_batches(test_x,
test_y, batch_size), 1):
            feed = {inputs_: x,
                    labels_: y[:, None],
                    keep_prob: 1,
                    initial_state: test_state}
            batch_acc, test_state =
    sess.run([accuracy, final_state],
    feed_dict=feed)
            test_acc.append(batch_acc)
        print("Test accuracy:
{:.3f}".format(np.mean(test_acc)))
```

Після декількох випробувань було виявлено, що десь після 20 епохи проходить перенавчання нейронної мережі і точність передбачень перестає зростати, тому було прийнято рішення зупинитися на 15 епохах.

Результати тестування зображено на рисунку 3.7.

```
>>> print("Test accuracy: {:.3f}".format(test_acc))
Test accuracy: 0.884
```

Рисунок 3.7 - Результати тестування для LSTM-мережі

Як можна побачити з результатів, застосування LSTM-мережі дало результат трохи кращий, ніж застосування простої згорткової мережі.

3.4 Навчання нейромережі

У якості попередньо навченої моделі буде використовуватися Google’s BERT. Оскільки модель вже навчена, в даному випадку нам не треба власноруч проводити тренування нейронної мережі, і модель одразу можна використовувати для виявлення тональності текстів.

Для того, щоб використовувати попередньо навчену модель, потрібно лише завантажити її з офіційного гітхаб-репозиторія.

Модель потребує набір з двох колонок, одна з яких – власне відгук, а друга – його полярність. Також треба задати вхідні параметри:

```
myparam =
    {
          "DATA_COLUMN": "text",
          "LABEL_COLUMN": "sentiment",
          "LEARNING_RATE": 2e-5,
          "NUM_TRAIN_EPOCHS":20
     }
```

Перші два параметри – це назви колонок у наборі даних для тестування. LEARNING_RATE залишимо за замовчуванням, а кількість епох вкажемо таку ж саму, як для попередніх досліджень.

Як і для минулих досліджень, треба провести токенізацію даних: кожне слово в відгуці представити у вигляді числа, після чого представити відгуки у вигляді векторів, так само, я у минулих дослідженнях. Для токенізації даних було використано такі ж самі методи, як для дослідження використання LSTM нейронної мережі.

Лейбли, що вказують на полярність тексту (позитивна чи негативна) знову мають бути представлені у вигляді бінарного флагу 0 чи 1.

Для того, щоб отримати результат, треба лише викликати функцію run_on_dfs, куди передається набір даних та визначені параметри. Результати виконання визначення полярності тексту зображено на рисунку 3.8.

Як можна побачити з результатів, точність визначення полярності трохи нижча, ніж для попередніх методів. Це пояснюється тим, що використану модель було попередньо навчено на іншому датасеті, і нейронною мережею не було виявлено ознаки, специфічні для власне відгуків на фільми.

```
{'auc': 0.856,
 'eval_accuracy': 0.856,
 'f1_score': 0.852459,
 'false_negatives': 84.0,
 'false_positives': 60.0,
 'global_step': 187,
 'loss': 0.530802,
 'precision': 0.8739496,
 'recall': 0.832,
 'true_negatives': 440.0,
 'true_positives': 416.0}
```

Рисунок 3.8 - Результати аналізу полярності відгуків з BERT

3.5 Розробка користуваційого інтерфейсу програми

Для бінарної класифікації тональності тексту, довільно заданого користувачем, було використано три попередньо треновані нейронні мережі: згорткова, LSTM-мережа та модель Google’s BERT.

Додаток реалізовано у вигляді веб-сторінки з полем для вводу текста та dropdown-елементом для вибору конкретної моделі (якщо жодна конкретна модель не була вибрана, то користувачеві видається результат для усіх трьох варіантів (див. рис. 3.9).

Користувацький інтерфейс побудовано з використанням jacascript-бібліотеки ReactJS, що дозволяє швидко збудувати клієнтський додаток і дає можливість відправляти запити на сервер.

Для проведення дослідження було обрано не лише відгуки на фільми з бази IMDB, а також випадкові тексти з наступних датасетів:

– Twitter US Airline Sentiment – цей набір даних містить дані Twitter про авіакомпанії США, які були зібрані з лютого 2015 року. Автори класифікували твіти як позитивні, негативні та нейтральні твіти (будуть використані лише позитивні та негативні);

– Paper Reviews – у цьому наборі даних містяться висловлені у наукових публікаціях речення, які позначають позитивні відгуки від міжнародної конференції з обчислювальної техніки та інформатики;

– Amazon Reviews for Sentiment Analysis – цей набір даних складається з декількох мільйонів відгуків клієнтів Amazon (вхідний текст) і рейтингів зірок (вихідні позначки) для навчання способу підготовки швидкого тексту для аналізу настроїв. Оскільки додаток працює з бінарною класифікацією, відгуки с рейтингом 4-5 вважаються позитивними, а з рейтингом 1-3 – негативними. Датасет не містить нейтральних відгуків.

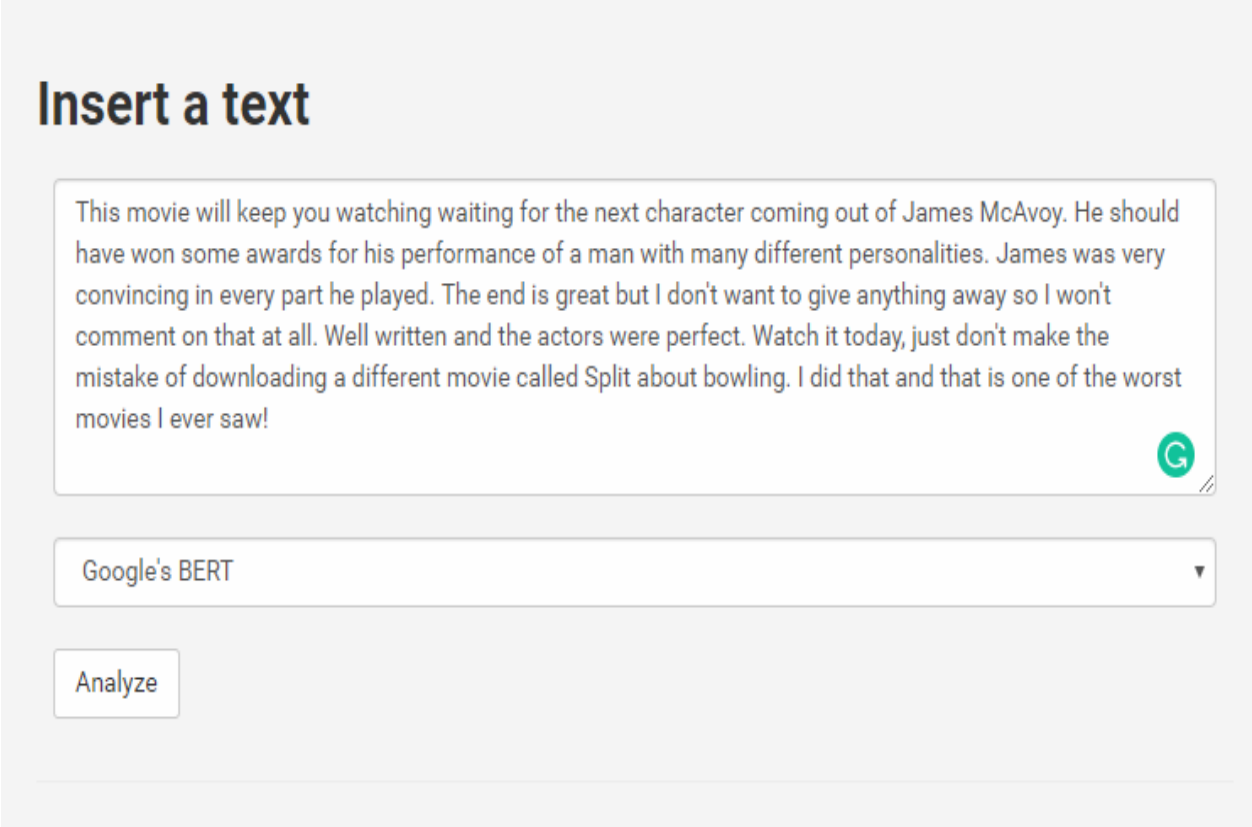


Рисунок 3.9 - Веб-інтерфейс для виявлення тональності тексту

3.6 Висновки

У третьому розділі було обгрунтовано вибір засоби реалізації програмного продукту. Було розроблено модуль машинного навчання із застосуванням згортковвих нейронних мереж. Також було розроблено моудь ідентифікації тональності тексту із застосуванням LTSM нейронної мережі. Розроблено метод та моделі системи визначення тональностей текстів Було розробено зручний і інтуїтивно зрозумістий користувацький інтерфейс програми.

4 ТЕСТУВАННЯ ДОДАТКУ

4.1 Визначення інструментів і даних для тестування додатку

Для проведення тестування додатку обрано набір даних, що містить 50000 відгуків на фільми з бази IMDB. Відгуки в цьому наборі даних промарковані як позитивні чи негативні, що дозволяє проводити бінарний аналіз тональності текстів. Відгуки написані англійською мовою.

Приклад позитивного відгуку до фільму “Титанік”:

“Why do people bitch about this movie and not about awful movies like The Godfather. Titanic is the greatest movie of the 21st Century.With great acting,directing,effects,music and generally everything. This movie is always dumped by all because one day some one said they didn't like it any more so most of the world decided to agree. There is nothing wrong with this movie. All I can say is that this movie, not only being the most heavily Oscar Awarded movie of all time, the most money ever made ever and sadly one of the most underrated movies I've ever seen. Apart from that it is truly the best movie of all time. The only movies that come close to being like all the Star Wars and the Lord of the Rings trilogy or anything by the masters Hitchcock or Spielberg or Tim Burton. These are all good movies and directors but none match up to James Cameron's Masterpiece TITANIC.”

Приклад негативного відгуку до фільму “Титанік”:

“1st watched 5/17/2002 - 3 out of 10(Dir-Ewald Andre Dupont): Fairly lame account of the Titanic disaster is the first filmed version of this much-heralded event. The replication of the disaster is not bad, but the drama around it is at some times silly, badly acted and way-too soap opera-like. The story is very much the same as the most recent Oscar-winning one except that we are shown how the crew tried to hide the actual disaster that was occurring until almost too late. Good for nostalgia purposes only and to get a feel for what James Cameron was competing against(barelyÂ…) in his recreation.”

Відгуки в наборі поділені на дві рівні частини: дані для тренування (25000 відгуків) та дані для тестування (25000 відгуків). Обидва набори містять рівну

кількість позитивних та негативних відгуків (рис. 4.1). Також набір даних додатково містить 50000 відгуків без інформації про забарвлення тексту.

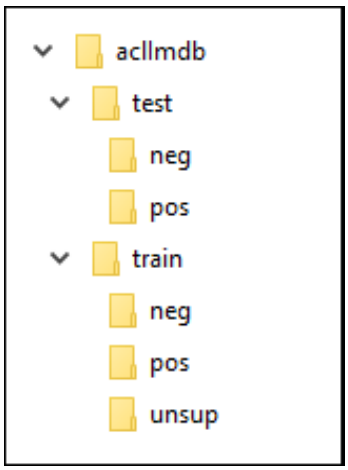


Рисунок 4.1 – Структура набору даних

4.2 Тестування модуля ідентифікації тональності з використанням згорткових нейронних мереж

Keras має вбудований доступ до датасету відгуків з сайту IMDB і може бути завантажений єдиною командою keras.datasets.imdb.load_data(). Формат датасету вже готовий для використання у нейронних мережах, тому можна пропустити етапи з конвертацією у lower case та видаленням знаків пунктуації. Щоб почати використовувати датасет, треба лише правильно вказати його назву та викликати вбудовану функцію Keras.

Навчання нейронної мережі проводиться за наступним алгоритмом:

– підраховується кількість використань кожного слова у датасеті;

– будується словник використаних слів, де ключом виступає порядковий номер слова, якщо відсортувати масив слів за кількістю використань; так, найбільш використаним є слово “this”;

– кожний відгук приводиться до довжини в 256 слів;

– кожний відгук конвертується в тензор;

– будується модель нейронної мережі;

– проводиться навчання нейронної мережі на тренувальному наборі даних;

– перевіряється результат роботи моделі на тестових даних.

Тренування моделі починається з 40 епох за допомогою міні-батчів по 512 зразків (Батч - набір, пакет даних). Це означає, зроблено 40 ітерацій (або проходів) по всім зразкам даних в тензори x_train і y_train (де x_train це власне відгуки до фільмів, представлені у вигляді векторів чисел, а y_train це вказання на те, відгук є позитивним чи негативним). Якщо модель має більше прихованих блоків, і/або більше шарів, то тоді нейросеть може навчитися більш складним уявленням.

Результати проходження навчання для перших десяти епох зображено на рисунку 4.2.

```
Epoch 1/40
15000/15000 [==============================] - 1s 81us/sample - loss: 0.6910 - acc: 0.6036 - val_loss: 0.6879 - val_acc: 0.6869
Epoch 2/40
15000/15000 [==============================] - 1s 68us/sample - loss: 0.6827 - acc: 0.7347 - val_loss: 0.6771 - val_acc: 0.7554
Epoch 3/40
15000/15000 [==============================] - 1s 72us/sample - loss: 0.6663 - acc: 0.7686 - val_loss: 0.6570 - val_acc: 0.7599
Epoch 4/40
15000/15000 [==============================] - 1s 70us/sample - loss: 0.6392 - acc: 0.7748 - val_loss: 0.6271 - val_acc: 0.7613
Epoch 5/40
15000/15000 [==============================] - 1s 69us/sample - loss: 0.6015 - acc: 0.8028 - val_loss: 0.5888 - val_acc: 0.7917
Epoch 6/40
15000/15000 [==============================] - 1s 67us/sample - loss: 0.5559 - acc: 0.8211 - val_loss: 0.5458 - val_acc: 0.8112
Epoch 7/40
15000/15000 [==============================] - 1s 70us/sample - loss: 0.5065 - acc: 0.8375 - val_loss: 0.5005 - val_acc: 0.8262
Epoch 8/40
15000/15000 [==============================] - 1s 69us/sample - loss: 0.4586 - acc: 0.8549 - val_loss: 0.4600 - val_acc: 0.8392
Epoch 9/40
15000/15000 [==============================] - 1s 71us/sample - loss: 0.4152 - acc: 0.8691 - val_loss: 0.4246 - val_acc: 0.8488
Epoch 10/40
15000/15000 [==============================] - 1s 69us/sample - loss: 0.3778 - acc: 0.8793 - val_loss: 0.3960 - val_acc: 0.8555
```

Рисунок 4.2 – Результати тренування нейронної мережі

Після навчання вимірюються втрати і точність нашої моделі шляхом перевірки на 25,000 зразків з перевірочного набору даних. Для перевірки моделі використовується функція evaluate, що отримує на вхід два масиви – тестові рецензії та відповідні маркування позитивної/негативної рецензії.

Результатом виконання цієї функції є пара чисел: відсоток втрат (loss; чим нижче це число, тим менше хибних прогнозів зробила нейронна мережа) та точність accuracy (див. рис. 4.3).

```
>>> print(results)
25000/25000 [==============================] - 1s 55us/sample - loss: 0.3346 - acc: 0.8699
[0.3346132341432, 0.8699]
```

Рисунок 4.3 – Результати роботи нейронної мережі на тестових даних.

Для більш наглядного відображення результатів побудовано графік втрат та точності для обох етапів – навчання та перевірки моделі за допомогою вбудованих можливостей використаних бібліотек.

На осі абсцис зображено епохи, на осі ординат – точність для кожної епохи. Крапками зображено точність при навчанні, а лінією – точність при перевірці моделі (див. рис. 4.4).

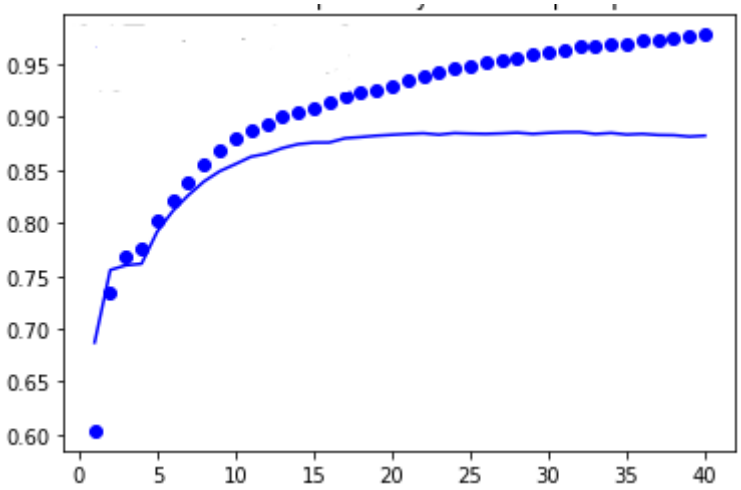


Рисунок 4.4 – Графік залежності точності від епох

Під час навчання втрати зменшуються, а точність збільшується з кожною наступною епохою. Але для перевірки моделі після двадцятої епохи точність перестає рости, що є ознакою перенавчання моделі – модель показує кращі результати на даних для навчання, аніж на нових даних для перевірки моделі. Тому має сенс зменшити кількість епох до 20.

Результати для 20 епох майже не відрізняються від результатів для 40 епох. Точність стала трохи більшою, але різниця незначна (див. рис. 4.5).

```
>>> print(results)
25000/25000 [==============================] - 1s 52us/sample - loss: 0.3365 - acc: 0.8706
[0.336505445151329, 0.8706]
```

Рисунок 4.5 – Результати для 20 епох

Отже, використання згорткової нейронної мережі для визначення тональності відгуків до фільмів показало досить прийнятний результат у 87% точності.

4.3 Тестування модуля ідентифікації тональності з застосування LTSM

Після декількох випробувань було виявлено, що десь після 20 епохи проходить перенавчання нейронної мережі і точність передбачень перестає зростати, тому було прийнято рішення зупинитися на 15 епохах.

Результати тестування зображено на рисунку 4.6.

```
>>> print("Test accuracy: {:.3f}".format(test_acc))
Test accuracy: 0.884
```

Рисунок 4.6 - Результати тестування для LSTM-мережі

Як можна побачити з результатів, застосування LSTM-мережі дало результат трохи кращий, ніж застосування простої згорткової мережі.

4.4 Тестування попередньо навченої мережі

У якості попередньо навченої моделі буде використовуватися Google’s BERT. Оскільки модель вже навчена, в даному випадку нам не треба власноруч проводити тренування нейронної мережі, і модель одразу можна використовувати для виявлення тональності текстів.

Для того, щоб використовувати попередньо навчену модель, потрібно лише завантажити її з офіційного гітхаб-репозиторія.

Як можна побачити з результатів, точність визначення полярності трохи нижча, ніж для попередніх методів. Це пояснюється тим, що використану модель було попередньо навчено на іншому датасеті, і нейронною мережею не було виявлено ознаки, специфічні для власне відгуків на фільми.

Для бінарної класифікації тональності тексту, довільно заданого користувачем, було використано три попередньо треновані нейронні мережі: згорткова, LSTM-мережа та модель Google’s BERT.

Для тестування додатку було обрано не лише відгуки на фільми з бази IMDB, а також випадкові тексти з наступних датасетів:

– Twitter US Airline Sentiment – цей набір даних містить дані Twitter про авіакомпанії США, які були зібрані з лютого 2015 року. Автори класифікували твіти як позитивні, негативні та нейтральні твіти (будуть використані лише позитивні та негативні);

– Paper Reviews – у цьому наборі даних містяться висловлені у наукових публікаціях речення, які позначають позитивні відгуки від міжнародної конференції з обчислювальної техніки та інформатики;

– Amazon Reviews for Sentiment Analysis – цей набір даних складається з декількох мільйонів відгуків клієнтів Amazon (вхідний текст) і рейтингів зірок (вихідні позначки) для навчання способу підготовки швидкого тексту для аналізу настроїв. Оскільки додаток працює з бінарною класифікацією, відгуки с рейтингом 4-5 вважаються позитивними, а з рейтингом 1-3 – негативними. Датасет не містить нейтральних відгуків.

Результати досліджень приведено у таблиці 4.1.

Таблиця 4.1 - Результати досліджень для різних датасетів

Датасет	Згорткова мережа	LSTM-мережа	BERT
Відгуки IMDB	0,86	0,87	0,85
Twitter US Airline Sentiment	0,67	0,701	0,83
Paper Reviews	0,76	0,79	0,84
Amazon Reviews	0,78	0,81	0,84

Як можна побачити з результатів дослідження, власні моделі краще спрацювали на тому ж датасеті, на якому проводилося тренування, в той час як для текстів іншої тематики власні треновані моделі показали гірший результат, що можна пояснити тим, що власні треновані моделі для тренування використовували менший корпус текстів, до того ж з однієї предметної області. Найгірший результат ці мережі показали у випадку з даними з твітера. Це можна пояснити тим, максимальна довжина повідомлення в твіттері - 280 знаків, в той час як середня довжина відгуку на IMDB - 240 слів.

Найкраще з новими датасетами впоралатся нейронна мережа BERT. Це можна пояснити тим, що вона натренована на менш специфічному датасеті та на більшому корпусі статей.

Однак при розширенні тренувального датасету для LSTM-мережі, потенційно можна досягти тих самих, або й навіть кращих результатів. Доцільним є використання корпусу даних різної довжини, а також з різних предметних областей.

4.5 Висновки

У четвертому розділі було визначено інструменти і данні, які використовувалися для тестування. Було протестовано модуль ідентифікації тональності з використанням згорткових нейронних мереж, який показав хороший результат (87%). Проведено тестування модуля ідентифікації тональності з застосуванням LTSM, який підтвердив працездатність роботи системи.

5 ЕКОНОМІЧНА ЧАСТИНА

5.1 Оцінювання комерційного потенціалу розробки

Метою проведення технологічного аудиту є оцінювання комерційного потенціалу розробки. Для проведення технологічного аудиту було залучено 2-х незалежних експертів. Такими експертами будуть Войтко В. В. та Крилик Л.В.

Здійснюємо оцінювання комерційного потенціалу розробки за 12-ма критеріями за 5-ти бальною шкалою.

Результати оцінювання комерційного потенціалу розробки наведено в таблиці 5.1.

Таблиця 5.1 – Результати оцінювання комерційного потенціалу розробки

Критерії	Прізвище, ініціали, посада експерта	
	1. Експерт 1	2. Експерт 2
	Бали, виставлені експертами:	
1	4	4
2	3	3
3	3	4
4	4	4
5	3	3
6	3	4
7	4	3
8	3	4
9	4	4
10	4	4
11	3	4
12	3	4
Сума балів	$СБ_1 = 42$	$СБ_2 = 45$
Середньоарифметична сума балів $\overline{СБ}$	$\overline{СБ} = \frac{\sum_{1}^{3} СБ_i}{2} = 43{,}5$	

Отже, з отриманих даних таблиці 5.1 видно, що нова розробка має високий рівень комерційного потенціалу.

5.2 Прогнозування витрат на виконання науково-дослідної роботи та конструкторсько–технологічної роботи

Для розробки нового програмного продукту необхідні такі витрати.

Основна заробітна плата для розробників визначається за формулою (5.1):

$$З_о = \frac{М}{Т_р} \cdot t, \quad (5.1)$$

де М- місячний посадовий оклад конкретного розробника;

$Т_р$ - кількість робочих днів у місяці, $Т_р = 21$ день;

t - число днів роботи розробника, t = 40 днів.

Розрахунки заробітних плат для керівника і програміста наведені в табл.5.2.

Таблиця 5.2 – Розрахунки основної заробітної плати

Працівник	Оклад М, грн.	Оплата за робочий день, грн.	Число днів роботи, t	Витрати на оплату праці, грн.
Науковий керівник	6500	309,52	8	2476,16
Інженер-програміст	4000	190,47	40	7618,8
Всього:				10094,16

Розрахуємо додаткову заробітну плату:

$$З_{дод} = 0,1 \cdot 10094,16 = 1009,41 \text{ (грн.)}$$

Нарахування на заробітну плату операторів НЗП розраховується як 37,5…40% від суми їхньої основної та додаткової заробітної плати (5.2):

$$Н_{зп}= (З_о + З_р)\cdot \frac{\beta}{100}, \quad (5.2)$$

$$Н_{зп}= (10094,16+1009,41)\cdot \frac{36,3}{100} = 4030,88 \text{ (грн.)}.$$

Розрахунок амортизаційних витрат для програмного забезпечення виконується за такою формулою (5.3):

$$А=\frac{Ц\cdot Н_а}{100}\cdot \frac{Т}{12}, \quad (5.3)$$

де Ц – балансова вартість обладнання, грн;

$Н_а$ – річна норма амортизаційних відрахувань % (для програмного забезпечення 25%);

Т – Термін використання (Т=3 міс.).

Таблиця 5.3 – Розрахунок амортизаційних відрахувань

Найменування програмного забезпечення	Балансова вартість, грн.	Норма амортизації, %	Термін використання, міс.	Величина амортизаційних відрахувань,грн
Персональний комп’ютер	9000	25	3	562,5
Всього:				562,5

Розрахуємо витрати на комплектуючі. Витрати на комплектуючі розрахуємо за формулою (5.4):

$$K = \sum_1^n Н_і \cdot Ц_і \cdot К_і, \quad (5.4)$$

де n – кількість комплектуючих;

Ні - кількість комплектуючих і-го виду;

Ці – покупна ціна комплектуючих і-го виду, грн;

Кі – коефіцієнт транспортних витрат (приймемо Кі = 1,1).

Таблиця 5.4 - Витрати на комплектуючі, що були використані для розробки ПЗ.

Найменування матеріалу	Одиниці виміру	Ціна, грн.	Витрачено	Вартість витрачених матеріалів, грн.
Флешка	шт.	150	1	150
Пачка паперу	уп.	100	1	100
Ручка	шт.	5	1	5
Всього з урахуванням транспортних витрат				280,5

Витрати на силову електроенергію розраховуються за формулою (5.5):

$$В_е=В\cdot П\cdot Ф\cdot К_п\ ; \quad (5.5)$$

де В – вартість 1кВт-години електроенергії (В=1,7 грн/кВт);

П – установлена потужність комп’ютера (П=0,6кВт);

Ф – фактична кількість годин роботи комп’ютера (Ф=195 год.);

$К_п$ – коефіцієнт використання потужності ($К_п< 1$, $К_п = 0{,}8$).

$$В_е= 1{,}7 \cdot 0{,}6 \cdot 195 \cdot 0{,}8 = 159{,}12 \text{ (грн.)}$$

Розрахуємо інші витрати $В_{ін}$.

Інші витрати $І_в$ можна прийняти як (100…300)% від суми основної заробітної плати розробників та робітників, які були виконували дану роботу, тобто (5.6):

$$В_{ін} = (1..3) \cdot (З_о + З_р). \quad (5.6)$$

Отже, розрахуємо інші витрати:

$$В_{ін} = 1 * (10094,16 + 1009,41) = 11103,57 \text{ (грн).}$$

Сума всіх попередніх статей витрат дає витрати на виконання даної частини роботи:

$$В = З_о + З_д + Н_{зп} + А + К + В_е + І_в$$

В = 10094,16+1009,41+4030,88+562,5+280,5+159,12+11103,57= 27240,14 (грн.)

Розрахуємо загальну вартість наукової роботи $B_{заг}$ за формулою (5.7):

$$В_{заг} = \frac{В_{ін}}{\alpha} \quad (5.7)$$

де α – частка витрат, які безпосередньо здійснює виконавець даного етапу роботи, у відн. одиницях = 1.

$$В_{заг} = \frac{27240,14}{1} = 27240,14$$

Прогнозування загальних витрат ЗВ на виконання та впровадження результатів виконаної наукової роботи здійснюється за формулою (5.8):

$$ЗВ = \frac{В_{заг}}{\beta} \quad (5.8)$$

де β – коефіцієнт, який характеризує етап (стадію) виконання даної роботи.

Отже, розрахуємо загальні витрати:

$$ЗВ = \frac{27240,14}{0,9} = 30266,82 \text{ (грн.)}$$

5.3 Прогнозування комерційних ефектів від реалізації результатів розробки

Спрогнозуємо отримання прибутку від реалізації результатів нашої розробки. Зростання чистого прибутку можна оцінити у теперішній вартості грошей. Це забезпечить підприємству (організації) надходження додаткових коштів, які дозволять покращити фінансові результати діяльності .

Оцінка зростання чистого прибутку підприємства від впровадження результатів наукової розробки. Збільшення чистого прибутку підприємства $\Delta П_i$ для кожного із років, протягом яких очікується отримання позитивних результатів від впровадження розробки, розраховується за формулою (5.9):

$$\Delta П_i = \sum_1^n (\Delta П_я \cdot N + П_я \Delta N)_i \quad (5.9)$$

де $\Delta П_я$ – покращення основного якісного показника від впровадження результатів розробки у даному році;

N – основний кількісний показник, який визначає діяльність підприємства у даному році до впровадження результатів наукової розробки;

ΔN – покращення основного кількісного показника діяльності підприємства від впровадження результатів розробки;

$П_я$ – основний якісний показник, який визначає діяльність підприємства у даному році після впровадження результатів наукової розробки;

n – кількість років, протягом яких очікується отримання позитивних результатів від впровадження розробки.

В результаті впровадження результатів наукової розробки витрати на виготовлення програмного продукту зменшаться на 30 грн (що автоматично спричинить збільшення чистого прибутку підприємства на 30 грн), а кількість користувачів, які будуть користуватись збільшиться: протягом першого року – на 150 користувачів, протягом другого року – на 100 користувачів, протягом третього року – 50 користувачів. Реалізація програмного продукту до впровадження результатів наукової розробки складала 1000 користувачів, а

прибуток, що отримував розробник до впровадження результатів наукової розробки – 300 грн.

Спрогнозуємо збільшення чистого прибутку від впровадження результатів наукової розробки у кожному році відносно базового.

Отже, збільшення чистого продукту $\Delta П_1$ протягом першого року складатиме:

$$\Delta П_1 = 30 \cdot 1000 + (300 + 30) \cdot 150 = 79500 \text{ грн.}$$

Протягом другого року:

$$\Delta П_2 = 30 \cdot 1000 + (300 + 30) \cdot (150 + 100) = 112500 \text{ грн.}$$

Протягом третього року:

$$\Delta П_3 = 30 \cdot 1000 + (300 + 30) \cdot (150 + 100 + 50) = 129000 \text{ грн.}$$

5.4 Розрахунок ефективності вкладених інвестицій та період їх окупності

Визначимо абсолютну і відносну ефективність вкладених інвестором інвестицій та розрахуємо термін окупності.

Абсолютна ефективність $Е_{абс}$вкладених інвестицій розраховується за формулою (5.10):

$$Е_{абс} = (ПП - PV), \qquad (5.10)$$

де $\Delta П_i$ – збільшення чистого прибутку у кожному із років, протягом яких виявляються результати виконаної та впровадженої НДДКР, грн;

т – період часу, протягом якого виявляються результати впровадженої НДДКР, 3 роки;

τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні; для України цей показник знаходиться на рівні 0,1;

t – період часу (в роках) від моменту отримання чистого прибутку до точки 2, 3,4.

Рисунок, що характеризує рух платежів (інвестицій та додаткових прибутків) буде мати вигляд, рисунок 5.1.

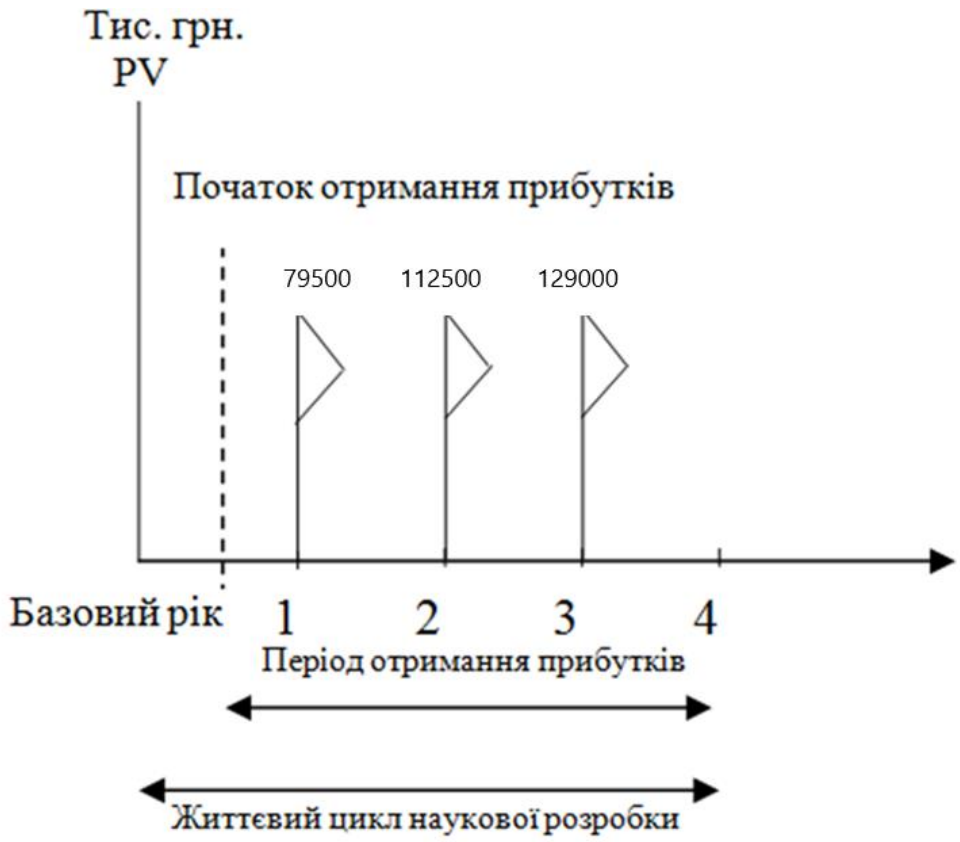


Рисунок 5.1 – Вісь часу з фіксацією платежів, що мають місце під час розробки та впровадження результатів НДДКР

Розрахуємо вартість чистих прибутків за формулою (5.11):

$$ПП = \sum_1^m \frac{\Delta П_i}{(1+\tau)^t} \quad (5.11)$$

де $\Delta П_i$ – збільшення чистого прибутку у кожному із років, протягом яких виявляються результати виконаної та впровадженої НДДКР, грн;

т – період часу, протягом якого виявляються результати впровадженої НДДКР, роки;

τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні; для України цей показник знаходиться на рівні 0,1;

t – період часу (в роках) від моменту отримання чистого прибутку до точки.

Отже, розрахуємо вартість чистого прибутку:

$$ПП = \frac{30266,82}{(1+0,1)^0} + \frac{79500}{(1+0,1)^2} + \frac{112500}{(1+0,1)^3} + \frac{129000}{(1+0,1)^4} = 268600,93 \text{ (грн.)}$$

Тоді розрахуємо $E_{абс}$:

$$E_{абс} = 268600,93 - 30266,82 = 238334,11 \text{ грн.}$$

Оскільки $E_{абс} > 0$, то вкладання коштів на виконання та впровадження результатів НДДКР буде доцільним.

Розрахуємо відносну (щорічну) ефективність вкладених в наукову розробку інвестицій $E_в$ за формулою (5.12):

$$E_в = \sqrt[T]{1 + \frac{E_{абс}}{PV}} - 1 \qquad (5.12)$$

де $E_{абс}$ – абсолютна ефективність вкладених інвестицій, грн;

PV – теперішня вартість інвестицій PV = ЗВ, грн;

$T_ж$ – життєвий цикл наукової розробки, роки.

Тоді будемо мати:

$$E_в = \sqrt[3]{1 + \frac{238334,11}{30266,82}} - 1 = 1,07 \text{ або } 107\ \%$$

Далі, розраховану величина $E_в$ порівнюємо з мінімальною (бар'єрною) ставкою дисконтування $\tau_{мін}$, яка визначає ту мінімальну дохідність, нижче за яку

інвестиції вкладатися не будуть. У загальному вигляді мінімальна (бар'єрна) ставка дисконтування $\tau_{мін}$ визначається за формулою:

$$\tau = d + f,$$

де d – середньозважена ставка за депозитними операціями в комерційних банках; в 2019 році в Україні d = 0,2;

f – показник, що характеризує ризикованість вкладень, величина f = 0,1.

$$\tau = 0{,}2 + 0{,}1 = 0{,}3$$

Оскільки $E_в$ = 107% > $\tau_{мін}$ = 0,3 = 30%, то у інвестор буде зацікавлений вкладати гроші в дану наукову розробку.

Термін окупності вкладених у реалізацію наукового проекту інвестицій. Термін окупності вкладених у реалізацію наукового проекту інвестицій $T_{ок}$ розраховується за формулою:

$$T_{ок} = \frac{1}{E_в}$$

$$T_{ок} = \frac{1}{1{,}07} = 0{,}93 \text{ року}$$

Обрахувавши термін окупності даної наукової розробки, можна зробити висновок, що фінансування даної наукової розробки буде доцільним.

5.5 Висновки

У пятому розділі було проведено оцінювання комерційного потенціалу розробки, виконано технологічний аудит для оцінювання комерційного потенціалу розробки. Залучено 2-х незалежних експертів. Спрогнозовано витрати на виконання науково-дослідної роботи та конструкторсько-технологічної роботи.

На вісі часу з фіксацією платежів, що мають місце під час розробки та впровадження результатів НДДКР показано життєвий цикл наукової розробки, початок отримання прибутків та його період.

ВИСНОВКИ

У магістерскій кваліфікаційній роботі подальшого розвитку дістав метод інтелектуального аналізу даних для визначення тональності тексту, який поєднує підходи словникового аналізу, блочної ідентифікації та машинного навчання і дозволяє підвищити об’єктивність результатів аналізу та забезпечив високу ефективність роботи системи. Розроблено моделі системи визначення тональності тексту, які орієнтовані на розробку та тренування спеціалізованої нейромережі, призначеної для лінгвістично-емоційного аналізу текстових документів, що підвищує реалістичність результатів ідентифікації текстових файлів.

Досліджено актуальність даної розробки. Було проаналізовано стан даної проблеми на сьогоднішній день.

Проведено варіантний аналіз засобів реалізації автоматизованої системи і обґрунтовано мови програмування Pithon.

Розроблено критерії автоматизованого визначення тональності тексту з використанням сучасних бібліотек та баз детасетів, які дозволяють реалізувати словниковий і блочний методи аналізу даних та забезпечити метод машинного навчання критеріальними обмеженнями. Розроблено програмне забезпечення неромережі та проведено навчання нейромережі.

Результати тестування створеного програмного продукту підтверджують його працездатність.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Bing Liu Sentiment Analysis: Mining Opinions, Sentiments and Emotion // Liu Bing. – Cambridge University Press; 1 edition, 2015 – 383 с.

2. Войтко В. В. Розробка засобів аналізу тональностей тексту / В.В.Войтко, С. В. Бевз, С.М. Бурбело, Л.М. Круподьорова, І.В. Кобися // Електронні інформаційні ресурси: створення, використання, доступ: Збірник матеріалів Міжнародної науково-практичної Інтернет-конференції. – Вінниця: ВНТУ, 2019. – с. 42-47.

3. Кобися І. В. Розробка засобів web-сайту «Вінниця – туристична перлина Поділля» / І. В. Кобися, Г. В. Богачук, А.В. Волошина, В. В. Войтко // XLVII Міжнародна науково-технічна конференція. – [Електронний ресурс]. – Режим доступу: https://conf.vntu.edu.ua/index.php/all-fitki/all-fitki-2018/paper/view/5196. – Назва з екрану.

4. Cambria E. Practical Guide to Sentiment Analysis / E. Cambria, D. Das, A. Bandyopadhyay. – Springer, 2017. – 199 с.

5. The best sentiment analysis tools / TalkWalker. – [Електронний ресурс]. – Режим доступу: https://www.talkwalker.com/blog/best-sentiment-analysis-tools. – Назва з екрану.

6. Blokdyk G. Sentiment Analysis a Complete Guide / G. Blokdyk // 5 starcooks, 2018. – 126 с.

7. Emotion and Sentiment Analysis: A Practitioner’s Guide to NLP. – [Електронний ресурс]. – Режим доступу: https://www.kdnuggets.com/2018/08/ emotion-sentiment-analysis-practitioners-guide-nlp-5.html. – Назва з екрану.

8. Passenger dragged off overbooked United flight. – [Електронний ресурс]. – Режим доступу: https://edition.cnn.com/2017/04/10/travel/passenger-removed-united-flight-trnd/ index.html. – Назва з екрану.

9. Hota S. KNN classifier based approach for multi-class sentiment analysis of twitter data / Soudamini Hota, Sudhir Pathak // Independently publisher, 2017. – 124с.

10. Sentiment Analysis: learn everything you need to know. – [Електронний ресурс]. – Режим доступу: https://monkeylearn.com/sentiment-analysis. – Назва з екрану.

11. Trump vs Hillary: Sentiment analysis on Twitter mentions. – [Електронний ресурс]. – Режим доступу: https://monkeylearn.com/blog/trump-vs-hillary-sentiment-analysis-twitter-mentions. – Назва з екрану.

12. IMDB Movies Review dataset. – [Електронний ресурс]. – Режим доступу: https://www.kaggle.com/iarunava/ imdb-movie-reviews-dataset. – Назва з екрану.

13. Chollet F. Deep Learning with Python Languagr / F. Chollet // 1st Edition. – 2017. – 384 с.

14. Pujari P. Practical Convolutional Neural Networks / Pradeep Pujari, Md. Rezaul Karim // Packt Publishing, 2018. – 218 с.

15. Millstein F. Deep Learning With Keras And Convolutional Neural Networks In Python Paperback / Frank Millstein, Deep Learning // 2 Manuscripts, 2018. – 260 с.

16. Bianchi F. M. Recurrent Neural Networks for Short-Term Load Forecasting / F. M. Bianchi, E. Maiorino // Springer Briefs, 2017. – 72 с.

17. Millstein F. Python Machine Learning: Introduction To Machine Learning With Python / F. Millstein // Kindle Edition,– 2018. – 134 с.

18. Kostadinov S. Recurrent Neural Networks with Python Quick Start Guide: Sequential learning and language modeling with Tensor Flow / Simeon Kostadinov // Paperback, 2018. – 122 с.

19. Chambers J. Recurrent Neural Networks for Prediction: Learning Algorithms / Jonathon Chambers // Architectures and Stability, Leicester. – 2011. – 105 с.

20. Lazy P. Deep Learning: Recurrent Neural Networks in Python: LSTM, GRU, and more RNN machine learning architectures in Python and Theano (Machine Learning in Python) / P. Lazy //. – Kindle Edition, 2016. – 56 с.

21. Майерс Г. Искусство тестирования программ / Гленфорд Майерс // The Art of Software Testing : Glenford J. Myers. – 272 с.

22. Transfer Learning / University of WISCONSIN. – [Електронний ресурс]. – Режим доступу: http://pages.cs.wisc.edu/~shavlik/abstracts/torrey. handbook09.abstract.html. – Назва з екрану.

23. BERT Explained: State of the art language model for NLP / Towards Data science. – [Електронний ресурс]. – Режим доступу: https://towardsdatascience.com/ bert-explained-state-of-the-art-language-model-for-nlp-f8b21a9b6270. – Назва з екрану.

ДОДАТКИ

Додаток А. Технічне завдання

Міністерство освіти і науки України
Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії

ЗАТВЕРДЖУЮ
д.т.н., проф. О. Н. Романюк
"____ "_____2019 р.

Технічне завдання
на магістерську кваліфікаційну роботу
«Розробка методу і програмних засобів аналізу тональності тексту з використанням нейронних мереж та веб-технологій»
за спеціальністю
121 – Інженерія програмного забезпечення

Керівник магістерської кваліфікаційної роботи:
_____ к.т.н., доцент кафедри ПЗ, В.В. Войтко
"____"_____2019 р.
Виконав:
_____студент гр. 1ПІ-18м, І.В. Кобися
"____"_____2019 р.

Вінниця ВНТУ 2019

1. Найменування та галузь застосування

Магістерська кваліфікаційна робота: «Розробка методу і програмних засобів аналізу тональності тексту з використанням нейронних мереж та веб-технологій».

Галузь застосування - системи комп'ютерної графіки.

2. Підстава для розробки.

Підставою для виконання магістерської кваліфікаційної роботи (МКР) є індивідуальне завдання на МКР та наказ № ректора по ВНТУ про закріплення тем МКР.

3. Мета та призначення розробки.

Мета дослідження – підвищення рівня реалістичності аналізу та ідентифікації текстових документів за рахунок розробки засобів визначення тональності тексту, що дозволяють автоматизувати процеси обробки даних.

Призначення роботи – розробка методу та системи автоматизованого визначення тональностей текстових ресурсів.

4. Вихідні дані для проведення НДР

Перелік основних літературних джерел, на основі яких буде виконуватись МКР.

1. Брайан Харди, Билл Филлипс,Крис Стюарт, Кристин Марсикано. Программирование под Android. 2-е издание (2016).
2. П. Дейтел, Х. Дейтел, А. Уолд. Android для разработчиков. 3-е издание(2016).
3. Блинов, Романчик. Java. Методы программирования.– СПб. : Минск, 2013. – 897 с.

5. Технічні вимоги

Мова програмування: Pyton

Технологія розробки: Веб-розробка

Браузери (або ОС): Windows, Chrome

6. Конструктивні вимоги.

Веб-додаток повинен відповідати всім вимогам, повинен бути зручним та зрозумілим у використанні.

Графічна та текстова документація повинна відповідати діючим стандартам України.

7. Перелік технічної документації, що пред'являється по закінченню робіт:

– пояснювальна записка до МКР;

– технічне завдання;

– лістинги програми.

8. Вимоги до рівня уніфікації та стандартизації

При розробці програмних засобів слід дотримуватися уніфікації і ДСТУ.

9. Стадії та етапи розробки:

№ з/п	Назва етапів магістерської кваліфікаційної Роботи	Строк виконання етапів роботи
1	Техніко-економічне обґрунтування доцільності розробки додатку для аналізу тональності тексту	07.10.2019 – 27.10-2019
2	Розробка модулів додатку для автоматизованого аналізу тональності тексту	28.10.2019 – 8.11.2019
3	Програмна реалізація додатку	9.11.2019 – 20.11.2019
4	Тестування роботи додатку	21.11.2019 – 3.12.2019
5	Економічне обґрунтування розробки програмного продукту	4.12.2019 – 7.12.2019

10. Порядок контролю та прийняття.

Виконання етапів магістерської кваліфікаційної роботи контролюється керівником згідно з графіком виконання роботи.

Прийняття магістерської кваліфікаційної роботи здійснюється ДЕК, затвердженою зав. кафедрою згідно з графіком.

Додаток Б

Програмний код модулів системи визначення тональностей текстів

Б.1 Згоркові нейронні мережі

```
    # keras.datasets.imdb is broken in 1.13 and 1.14, by
np 1.16.3
    !pip install tf_nightly

    from __future__ import absolute_import, division,
print_function, unicode_literals

    import tensorflow as tf
    from tensorflow import keras

    import numpy as np

    print(tf.__version__)

    imdb = keras.datasets.imdb

    (train_data, train_labels), (test_data, test_labels) =
imdb.load_data(num_words=10000)

    print("Training entries: {}, labels:
{}".format(len(train_data), len(train_labels)))

    print(train_data[0])

    print(train_data[0])

    # A dictionary mapping words to an integer index
    word_index = imdb.get_word_index()

    # The first indices are reserved
    word_index = {k:(v+3) for k,v in word_index.items()}
    word_index["<PAD>"] = 0
    word_index["<START>"] = 1
    word_index["<UNK>"] = 2  # unknown
    word_index["<UNUSED>"] = 3

    reverse_word_index = dict([(value, key) for (key,
value) in word_index.items()])
```

```
    def decode_review(text):
        return  '  '.join([reverse_word_index.get(i,  '?')
for i in text])


    train_data                                              =
keras.preprocessing.sequence.pad_sequences(train_data,

value=word_index["<PAD>"],

padding='post',

maxlen=256)

    test_data                                               =
keras.preprocessing.sequence.pad_sequences(test_data,

value=word_index["<PAD>"],

padding='post',

maxlen=256)

    # input shape is the vocabulary count used for the
movie reviews (10,000 words)
    vocab_size = 10000

    model = keras.Sequential()
    model.add(keras.layers.Embedding(vocab_size, 16))
    model.add(keras.layers.GlobalAveragePooling1D())
    model.add(keras.layers.Dense(16,
activation=tf.nn.relu))
    model.add(keras.layers.Dense(1,
activation=tf.nn.sigmoid))

    model.summary()

    # input shape is the vocabulary count used for the
movie reviews (10,000 words)
    vocab_size = 10000

    model = keras.Sequential()
    model.add(keras.layers.Embedding(vocab_size, 16))
    model.add(keras.layers.GlobalAveragePooling1D())
```

```
    model.add(keras.layers.Dense(16,
activation=tf.nn.relu))
    model.add(keras.layers.Dense(1,
activation=tf.nn.sigmoid))

    model.summary()

    # input shape is the vocabulary count used for the
movie reviews (10,000 words)
    vocab_size = 10000

    model = keras.Sequential()
    model.add(keras.layers.Embedding(vocab_size, 16))
    model.add(keras.layers.GlobalAveragePooling1D())
    model.add(keras.layers.Dense(16,
activation=tf.nn.relu))
    model.add(keras.layers.Dense(1,
activation=tf.nn.sigmoid))

    model.summary()

    history = model.fit(partial_x_train,
                        partial_y_train,
                        epochs=40,
                        batch_size=512,
                        validation_data=(x_val, y_val),
                        verbose=1)

    results = model.evaluate(test_data, test_labels)

    print(results)


    history_dict = history.history
    history_dict.keys()

    import matplotlib.pyplot as plt

    acc = history_dict['acc']
    val_acc = history_dict['val_acc']
    loss = history_dict['loss']
    val_loss = history_dict['val_loss']

    epochs = range(1, len(acc) + 1)
```

```
    # "bo" is for "blue dot"
    plt.plot(epochs, loss, 'bo', label='Training loss')
    # b is for "solid blue line"
    plt.plot(epochs,   val_loss,   'b',   label='Validation
loss')
    plt.title('Training and validation loss')
    plt.xlabel('Epochs')
    plt.ylabel('Loss')
    plt.legend()

    plt.show()

    plt.clf()   # clear figure

    plt.plot(epochs, acc, 'bo', label='Training acc')
    plt.plot(epochs, val_acc, 'b', label='Validation acc')
    plt.title('Training and validation accuracy')
    plt.xlabel('Epochs')
    plt.ylabel('Accuracy')
    plt.legend()

    plt.show()
```

Б.2 Рекурентні нейронні мережі

```
    import torch.nn as nn

    class SentimentLSTM(nn.Module):
        """
        The RNN model that will be used to perform
Sentiment analysis.
        """

        def __init__(self, vocab_size, output_size,
embedding_dim, hidden_dim, n_layers, drop_prob=0.5):
            """
            Initialize the model by setting up the layers.
            """
            super().__init__()

            self.output_size = output_size
```

```
            self.n_layers = n_layers
            self.hidden_dim = hidden_dim

            # embedding and LSTM layers
            self.embedding    =    nn.Embedding(vocab_size,
embedding_dim)
            self.lstm = nn.LSTM(embedding_dim, hidden_dim,
n_layers,
                                dropout=drop_prob,
batch_first=True)

            # dropout layer
            self.dropout = nn.Dropout(0.3)

            # linear and sigmoid layers
            self.fc = nn.Linear(hidden_dim, output_size)
            self.sig = nn.Sigmoid()


        def forward(self, x, hidden):
            """
            Perform a forward pass of our model on some
input and hidden state.
            """
            batch_size = x.size(0)

            # embeddings and lstm_out
            embeds = self.embedding(x)
            lstm_out, hidden = self.lstm(embeds, hidden)

            # stack up lstm outputs
            lstm_out    =    lstm_out.contiguous().view(-1,
self.hidden_dim)

            # dropout and fully-connected layer
            out = self.dropout(lstm_out)
            out = self.fc(out)
            # sigmoid function
            sig_out = self.sig(out)

            # reshape to be batch_size first
            sig_out = sig_out.view(batch_size, -1)
            sig_out = sig_out[:, -1] # get last batch of
labels
```

```
            # return last sigmoid output and hidden state
            return sig_out, hidden


        def init_hidden(self, batch_size):
            ''' Initializes hidden state '''
            # Create two new tensors with sizes n_layers x
batch_size x hidden_dim,
            # initialized to zero, for hidden state and
cell state of LSTM
            weight = next(self.parameters()).data

            if (train_on_gpu):
                hidden     =     (weight.new(self.n_layers,
batch_size, self.hidden_dim).zero_().cuda(),
                       weight.new(self.n_layers,
batch_size, self.hidden_dim).zero_().cuda())
            else:
                hidden     =     (weight.new(self.n_layers,
batch_size, self.hidden_dim).zero_(),
                           weight.new(self.n_layers,
batch_size, self.hidden_dim).zero_())

            return hidden

    # loss and optimization functions
    lr=0.001

    criterion = nn.BCELoss()
    optimizer = torch.optim.Adam(net.parameters(), lr=lr)


    # training params

    epochs = 4 # 3-4 is approx where I noticed the
validation loss stop decreasing

    counter = 0
    print_every = 100
    clip=5 # gradient clipping

    # move model to GPU, if available
    if(train_on_gpu):
        net.cuda()
```

```
    net.train()
    # train for some number of epochs
    for e in range(epochs):
        # initialize hidden state
        h = net.init_hidden(batch_size)

        # batch loop
        for inputs, labels in train_loader:
            counter += 1

            if(train_on_gpu):
                inputs,     labels     =     inputs.cuda(),
labels.cuda()

            # Creating new variables for the hidden state,
otherwise
            #  we'd backprop through the entire training
history
            h = tuple([each.data for each in h])

            # zero accumulated gradients
            net.zero_grad()

            # get the output from the model
            inputs = inputs.type(torch.LongTensor)
            output, h = net(inputs, h)

            # calculate the loss and perform backprop
            loss       =       criterion(output.squeeze(),
labels.float())
            loss.backward()
            # `clip_grad_norm` helps prevent the exploding
gradient problem in RNNs / LSTMs.
            nn.utils.clip_grad_norm_(net.parameters(),
clip)
            optimizer.step()

            # loss stats
            if counter % print_every == 0:
                # Get validation loss
                val_h = net.init_hidden(batch_size)
                val_losses = []
                net.eval()
                for inputs, labels in valid_loader:
```

```
                    # Creating new variables for the
hidden state, otherwise
                    # we'd backprop through the entire
training history
                    val_h = tuple([each.data for each in
val_h])

                    if(train_on_gpu):
                        inputs, labels = inputs.cuda(),
labels.cuda()

                    inputs = inputs.type(torch.LongTensor)
                    output, val_h = net(inputs, val_h)
                    val_loss = criterion(output.squeeze(),
labels.float())

                    val_losses.append(val_loss.item())

                net.train()
                print("Epoch:         {}/{}...".format(e+1,
epochs),
                      "Step: {}...".format(counter),
                      "Loss:
{:.6f}...".format(loss.item()),
                      "Val                              Loss:
{:.6f}".format(np.mean(val_losses)))
```

Додаток В

Користувацький інтерфейс додатку для аналізу тональності тексту